Virtex-6 FPGA Configurable Logic Block

User Guide

UG364 (v1.1) September 16, 2009

XILINX®

Xilinx is disclosing this user guide, manual, release note, and/or specification (the "Documentation") to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© 2009 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/24/09	1.0	Initial Xilinx release.
09/16/09	1.1	Add Virtex-6 HXT devices to Table 2. Updated discussions at Look-Up Table (LUT), page 11, and Static Read Operation, page 29. CLB labeling change in figures throughout document (Figure 6 through Figure 13, Figure 15, Figure 17, Figure 27 through Figure 29), including clarifying the T_{DS}/T_{DH} functions, descriptions, and notes in Table 7, page 39 and Table 8, page 42.

Table of Contents

Revision History 2

Preface: About This Guide

Additional Documentation 5
Additional Support Resources 6

Virtex-6 FPGA CLB

CLB Overview 7
Slice Description 8
CLB / Slice Timing Models 34
General Slice Timing Model and Parameters 34
Slice Distributed RAM Timing Model and Parameters (Available in SLICEM only) 38
Slice SRL Timing Model and Parameters (Available in SLICEM only) 41
Slice Carry-Chain Timing Model and Parameters 44
CLB Primitives 45
Distributed RAM Primitives 45
Shift Registers (SRLs) Primitive 47
Other Shift Register Applications 48
Multiplexer Primitives 49
Carry Chain Primitive 49

About This Guide

This guide serves as a technical reference describing the Virtex®-6 FPGA configurable logic blocks (CLBs). Usually, the logic synthesis software assigns the CLB resources without system designer intervention. It can be advantageous for the designer to understand certain CLB details, including the varying capabilities of the look-up tables (LUTs), the physical direction of the carry propagation, the number and distribution of the available flip-flops, and the availability of the very efficient shift registers. This guide describes these and other features of the CLB in detail.

Additional Documentation

The following documents are also available for download at http://www.xilinx.com/support/documentation/virtex-6.htm.

- Virtex-6 Family Overview

  The features and product selection of the Virtex-6 family are outlined in this overview.

- Virtex-6 FPGA Data Sheet: DC and Switching Characteristics

  This data sheet contains the DC and Switching Characteristic specifications for the Virtex-6 family.

- Virtex-6 FPGA Packaging and Pinout Specifications

  This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

- Virtex-6 FPGA Configuration Guide

  This all-encompassing configuration guide includes chapters on configuration interfaces (serial and SelectMAP), bitstream encryption, boundary-scan and JTAG configuration, reconfiguration techniques, and readback through the SelectMAP and JTAG interfaces.

- Virtex-6 FPGA Clocking Resources User Guide

  This guide describes the clocking resources available in all Virtex-6 devices, including the MMCM and PLLs.

- Virtex-6 FPGA Memory Resources User Guide

  The functionality of the block RAM and FIFO are described in this user guide.

- Virtex-6 FPGA SelectIO Resources User Guide

  This guide describes the SelectIO™ resources available in all Virtex-6 devices.

- Virtex-6 FPGA GTH Transceivers User Guide

  This guide describes the GTH transceivers available in all Virtex-6 HXT FPGAs except the XC6VHX250T and the XC6VHX380T in the FF1154 package..

- Virtex-6 FPGA GTX Transceivers User Guide

  This guide describes the GTX transceivers available in all Virtex-6 FPGAs except the XC6VLX760.

- Virtex-6 FPGA Embedded Tri-Mode Ethernet MAC User Guide

  This guide describes the dedicated Tri-Mode Ethernet Media Access Controller available in all Virtex-6 FPGAs except the XC6VLX760.

- Virtex-6 FPGA DSP48E1 Slice User Guide

  This guide describes the architecture of the DSP48E1 slice in Virtex-6 FPGAs and provides configuration examples.

- Virtex-6 FPGA System Monitor User Guide

  The System Monitor functionality available in all Virtex-6 devices is outlined in this guide.

- Virtex-6 FPGA PCB Design Guide

  This guide provides information on PCB design for Virtex-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

Additional Support Resources

To search the database of silicon and software questions and answers or to create a technical support case in WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

Virtex-6 FPGA CLB

CLB Overview

The Configurable Logic Blocks (CLBs) are the main logic resources for implementing sequential as well as combinatorial circuits. Each CLB element is connected to a switch matrix for access to the general routing matrix (shown in Figure 1). A CLB element contains a pair of slices. These two slices do not have direct connections to each other, and each slice is organized as a column. Each slice in a column has an independent carry chain. For each CLB, slices in the bottom of the CLB are labeled as SLICE(0), and slices in the top of the CLB are labeled as SLICE(1).

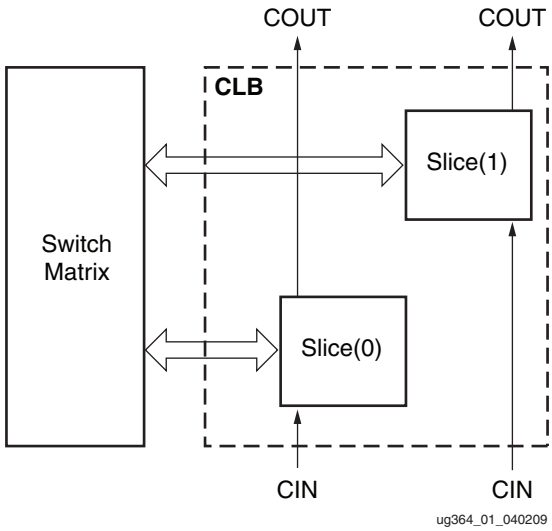

Figure 1: **Arrangement of Slices within the CLB**

The Xilinx tools designate slices with the following definitions. An "X" followed by a number identifies the position of each slice in a pair as well as the column position of the slice. The "X" number counts slices starting from the bottom in sequence 0, 1 (the first CLB column); 2, 3 (the second CLB column); etc. A "Y" followed by a number identifies a row of slices. The number remains the same within a CLB, but counts up in sequence from one CLB row to the next CLB row, starting from the bottom. Figure 2 shows four CLBs located in the bottom-left corner of the die.

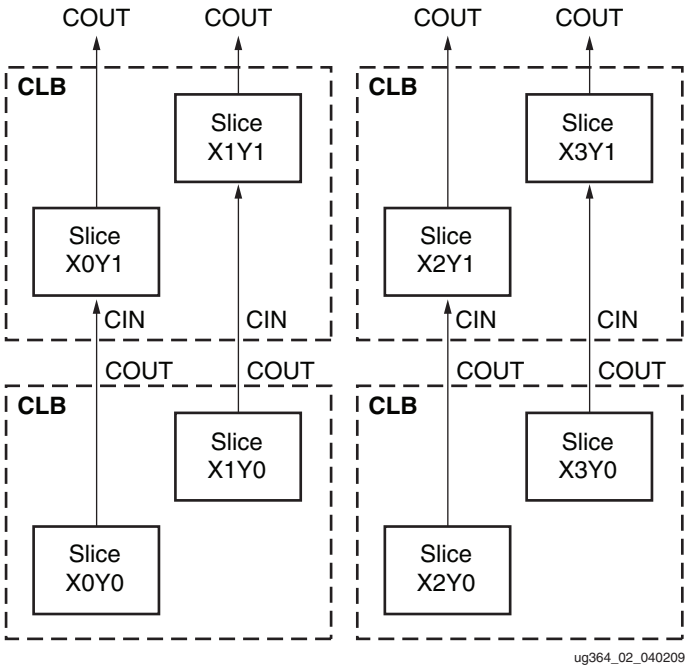

Figure 2: **Row and Column Relationship between CLBs and Slices**

Slice Description

Every slice contains four logic-function generators (or look-up tables), eight storage elements, wide-function multiplexers, and carry logic. These elements are used by all slices to provide logic, arithmetic, and ROM functions. In addition to this, some slices support two additional functions: storing data using distributed RAM and shifting data with 32-bit registers. Slices that support these additional functions are called SLICEM; others are called SLICEL. SLICEM (shown in Figure 3) represents a superset of elements and connections found in all slices. SLICEL is shown in Figure 4.

Each CLB can contain zero or one SLICEM. Every other CLB column contains a SLICEMs. In addition, the two CLB columns to the left of the DSP48E columns both contain a SLICEL and a SLICEM.

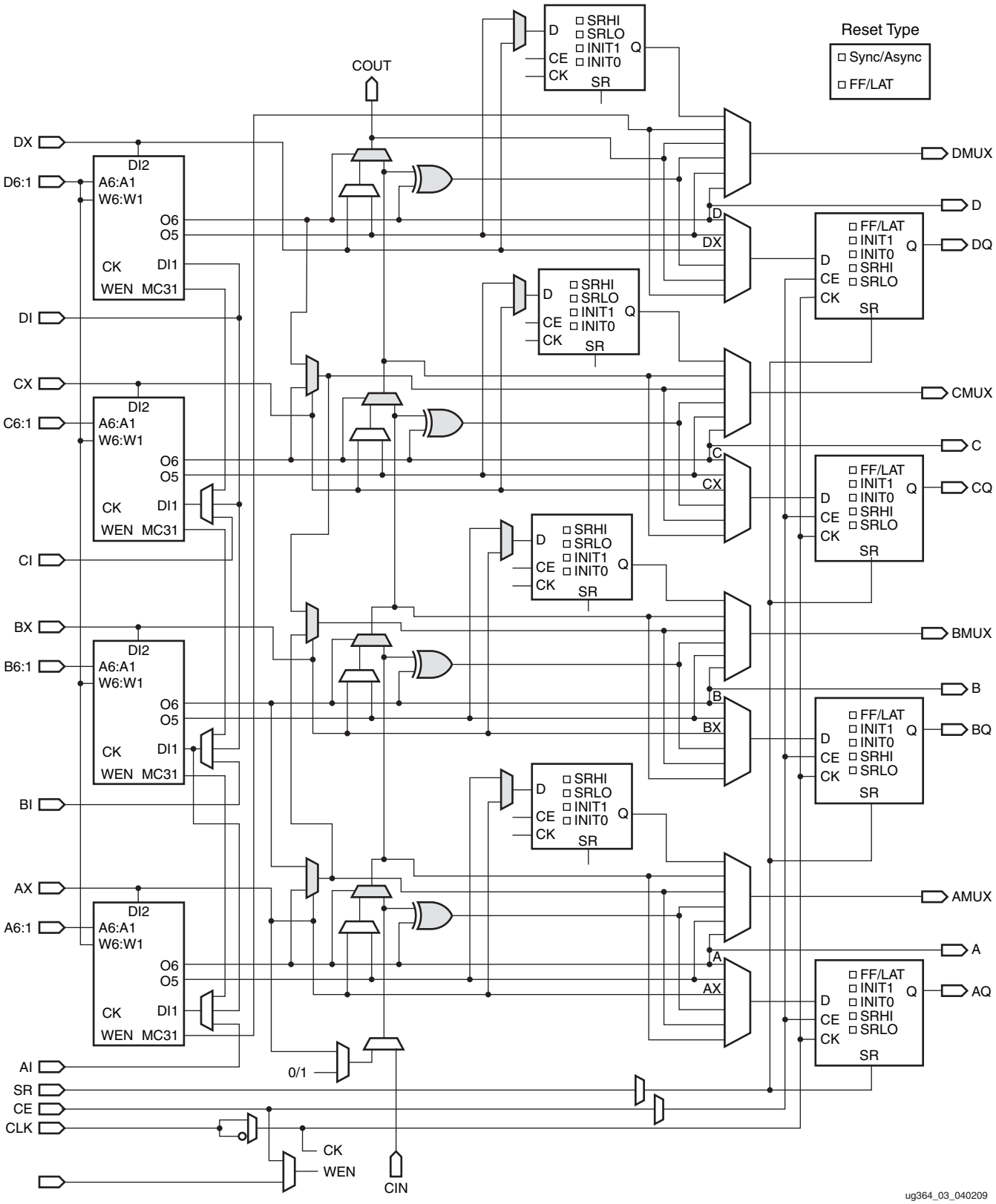

Figure 3: **Diagram of SLICEM**

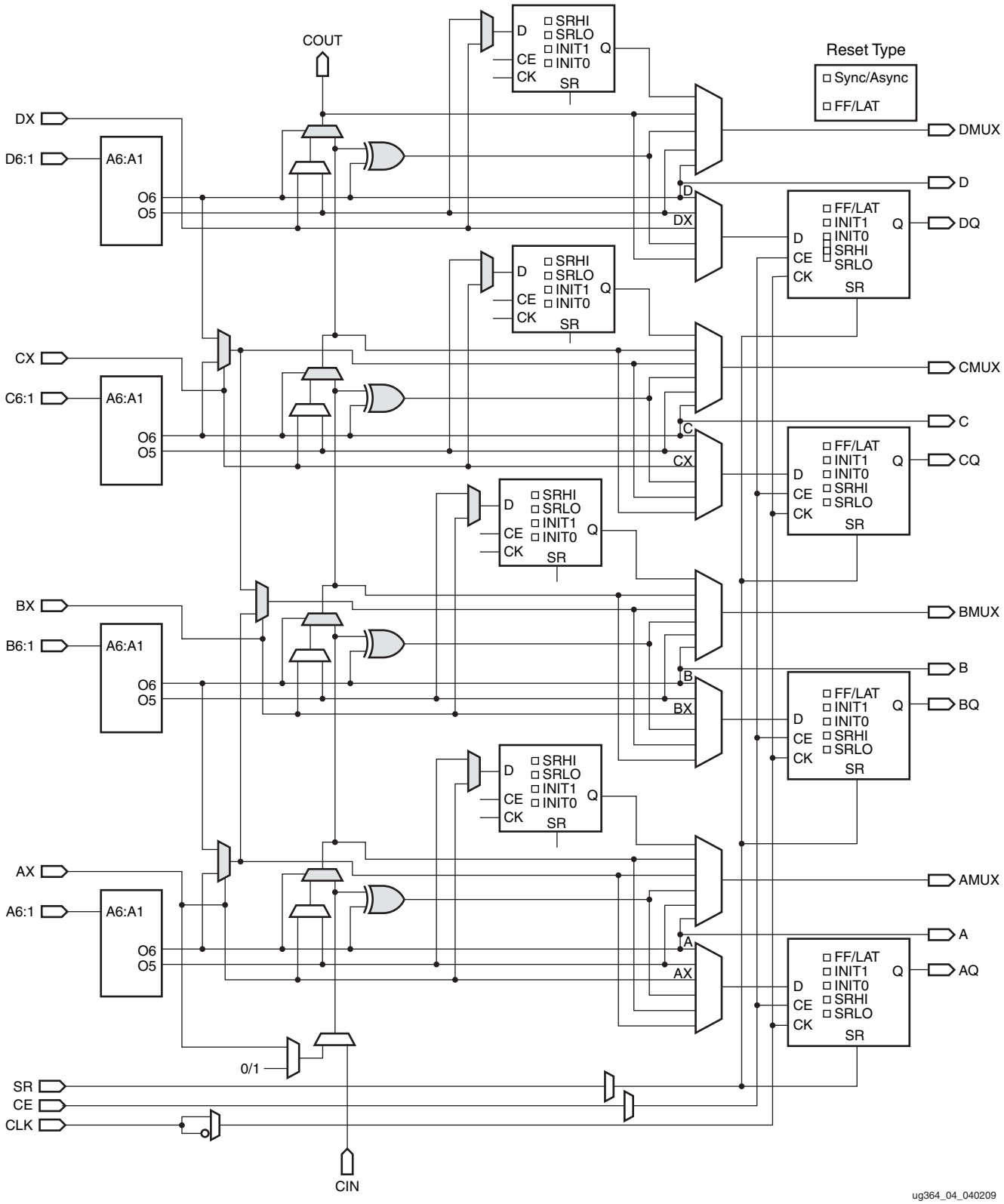

Figure 4: **Diagram of SLICEL**

CLB/Slice Configurations

Table 1 summarizes the logic resources in one CLB. Each CLB or slice can be implemented in one of the configurations listed. Table 2 shows the available resources in all CLBs.

Table 1: **Logic Resources in One CLB**

Slices	LUTs	Flip-Flops	Arithmetic and Carry Chains	Distributed RAM(1)	Shift Registers(1)
2	8	16	2	256 bits	128 bits

Notes:

1. SLICEM only, SLICEL does not have distributed RAM or shift registers.

Table 2: **Virtex-6 FPGA Logic Resources Available in All CLBs**

Device	Total Slices	SLICELs	SLICEMs	Number of 6-Input LUTs	Maximum Distributed RAM (Kb)	Shift Register (Kb)	Number of Flip-Flops
XC6VLX75T	11,640	7,460	4,180	46,560	1,045	522.5	93,120
XC6VLX130T	20,000	13,040	6,960	80,000	1,740	870	160,000
XC6VLX195T	31,200	19,040	12,160	124,800	3,140	1,570	249,600
XC6VLX240T	37,680	23,080	14,600	150,720	3,770	1,885	301,440
XC6VLX365T	56,880	40,360	16,520	227,520	4,130	2,065	455,040
XC6VLX550T	85,920	61,120	24,800	343,680	6,200	3,100	687,360
XC6VLX760	118,560	85,440	33,120	474,240	8,280	4,140	948,480
XC6VSX315T	49,200	28,840	20,360	196,800	5,090	2,545	393,600
XC6VSX475T	74,400	48,840	30,560	297,600	7,640	3,820	595,200
XC6VHX250T	39,360	27,200	12,160	157,440	3,040	1,520	314,880
XC6VHX255T	39,600	27,400	12,200	158,400	3,050	1,525	316,800
XC6VHX380T	59,760	41,520	18,240	239,040	4,570	2,285	478,080
XC6VHX565T	88,560	63,080	25,480	354,240	6,370	3,185	708,480

Look-Up Table (LUT)

The function generators in Virtex-6 FPGAs are implemented as six-input look-up tables (LUTs). There are six independent inputs (A inputs - A1 to A6) and two independent outputs (O5 and O6) for each of the four function generators in a slice (A, B, C, and D). The function generators can implement any arbitrarily defined six-input Boolean function. Each function generator can also implement two arbitrarily defined five-input Boolean functions, as long as these two functions share common inputs. Only the O6 output of the function generator is used when a six-input function is implemented. Both O5 and O6 are used for each of the five-input function generators implemented. In this case, A6 is driven High by the software. The propagation delay through a LUT is independent of the function implemented, or whether one six-input or two five-input generators are implemented. Signals from the function generators can exit the slice (through A, B, C, D output for O6 or AMUX, BMUX, CMUX, DMUX output for O5), enter the XOR dedicated gate from an O6 output (see Fast Lookahead Carry Logic), enter the carry-logic chain from an O5 output (see Fast Lookahead Carry Logic), enter the select line of the carry-logic multiplexer from

O6 output (see Fast Lookahead Carry Logic), feed the D input of the storage element, or go to F7AMUX/F7BMUX from O6 output.

In addition to the basic LUTs, slices contain three multiplexers (F7AMUX, F7BMUX, and F8MUX). These multiplexers are used to combine up to four function generators to provide any function of seven or eight inputs in a slice. F7AMUX and F7BMUX are used to generate seven input functions from LUTs A and B, or C and D, while F8MUX is used to combine all LUTs to generate eight input functions. Functions with more than eight inputs can be implemented using multiple slices. There are no direct connections between slices to form function generators greater than eight inputs within a CLB.

Storage Elements

As in previous Virtex architectures, there are four (original) storage elements in a slice that can be configured as either edge-triggered D-type flip-flops or level-sensitive latches. The D input can be driven directly by a LUT output via AFFMUX, BFFMUX, CFFMUX or DFFMUX, or by the BYPASS slice inputs bypassing the function generators via AX, BX, CX, or DX input. When configured as a latch, the latch is transparent when the CLK is Low.

In Virtex-6 devices, there are now four additional storage elements that can only be configured as edge-triggered D-type flip-flops. The D input can be driven by the O5 output of the LUT or the BYPASS slice inputs via AX, BX, CX, or DX input. When the original 4 storage elements are configured as latches, these 4 additional storage elements can not be used.

The control signals clock (CLK), clock enable (CE), and set/reset (SR) are common to all storage elements in one slice. When one flip-flop in a slice has SR or CE enabled, the other flip-flops used in the slice will also have SR or CE enabled by the common signal. Only the CLK signal has independent polarity. Any inverter placed on the clock signal is automatically absorbed. The CE and SR signals are active High. All flip-flop and latch primitives have CE and non-CE versions.

The SR signal forces the storage element into the state specified by the attribute SRHIGH or SRLOW. SRHIGH forces a logic High at the storage element output when SR is asserted, while SRLOW forces a logic Low at the storage element output (see Table 3).

Table 3: **Truth Table when using SRLOW and SRHIGH**

SR	SRVAL	Function
0	SRLOW (default)	No Logic Change
1	SRLOW (default)	0
0	SRHIGH	No Logic Change
1	SRHIGH	1

Figure 5 shows both the register only and the register/latch configuration in a slice.

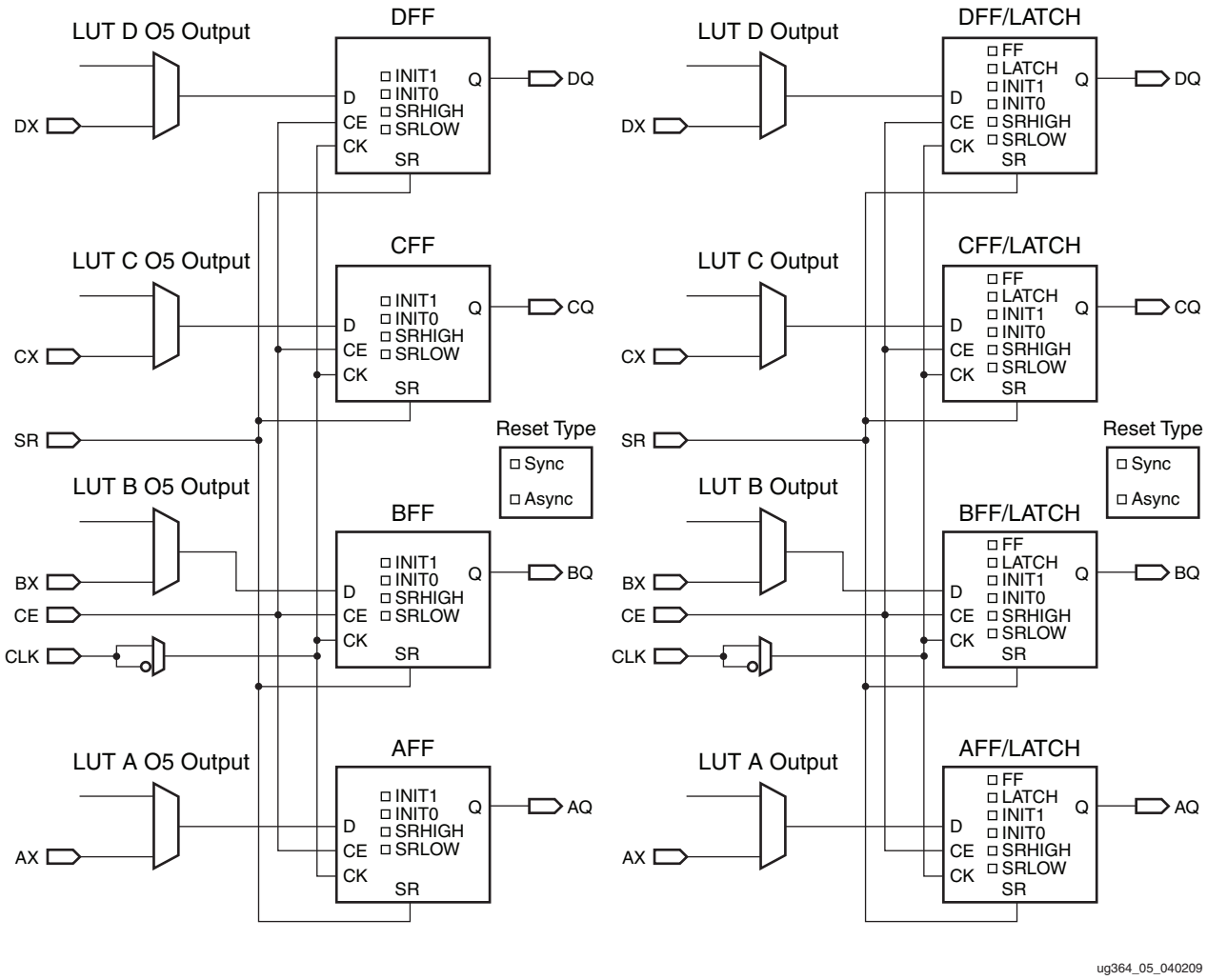

Figure 5: **Two Versions of Configuration in a Slice: 4 Registers Only and 4 Register/Latch**

SRHIGH and SRLOW can be set individually for each storage element in a slice. The choice of synchronous (SYNC) or asynchronous (ASYNC) set/reset (SRTYPE) cannot be set individually for each storage element in a slice.

The initial state after configuration or global initial state is defined by separate INIT0 and INIT1 attributes. By default, setting the SRLOW attribute sets INIT0, and setting the SRHIGH attribute sets INIT1. Virtex-6 devices can set INIT0 and INIT1 independent of SRHIGH and SRLOW.

The configuration options for the set and reset functionality of a register or the four storage elements capable of functioning as a latch are as follows:

- No set or reset
- Synchronous set
- Synchronous reset
- Asynchronous set (preset)
- Asynchronous reset (clear)

Distributed RAM and Memory (Available in SLICEM only)

Multiple LUTs in a SLICEM can be combined in various ways to store larger amount of data.

The function generators (LUTs) in SLICEMs can be implemented as a synchronous RAM resource called a distributed RAM element. RAM elements are configurable within a SLICEM to implement the following:

- Single-Port 32 x 1-bit RAM
- Dual-Port 32 x 1-bit RAM
- Quad-Port 32 x 2-bit RAM
- Simple Dual-Port 32 x 6-bit RAM
- Single-Port 64 x 1-bit RAM
- Dual-Port 64 x 1-bit RAM
- Quad-Port 64 x 1-bit RAM
- Simple Dual-Port 64 x 3-bit RAM
- Single-Port 128 x 1-bit RAM
- Dual-Port 128 x 1-bit RAM
- Single-Port 256 x 1-bit RAM

Distributed RAM modules are synchronous (write) resources. A synchronous read can be implemented with a storage element or a flip-flop in the same slice. By placing this flip-flop, the distributed RAM performance is improved by decreasing the delay into the clock-to-out value of the flip-flop. However, an additional clock latency is added. The distributed elements share the same clock input. For a write operation, the Write Enable (WE) input, driven by either the CE or WE pin of a SLICEM, must be set High.

Table 4 shows the number of LUTs (four per slice) occupied by each distributed RAM configuration.

Table 4: **Distributed RAM Configuration**

RAM	Number of LUTs
32 x 1S	1
32 x 1D	2
32 x 2Q(2)	4
32 x 6SDP(2)	4
64 x 1S	1
64 x 1D	2
64 x 1Q(3)	4
64 x 3SDP(3)	4
128 x 1S	2
128 x 1D	4

Table 4: **Distributed RAM Configuration *(Cont'd)***

RAM	Number of LUTs
256 x 1S	4

Notes:

1. S = single-port configuration; D = dual-port configuration; Q = quad-port configuration; SDP = simple dual-port configuration.
2. RAM32M is the associated primitive for this configuration.
3. RAM64M is the associated primitive for this configuration.

For single-port configurations, distributed RAM has a common address port for synchronous writes and asynchronous reads. For dual-port configurations, distributed RAM has one port for synchronous writes and asynchronous reads, and another port for asynchronous reads. In simple dual-port configuration, there is no data out (read port) from the write port. For quad-port configurations, distributed RAM has one port for synchronous writes and asynchronous reads, and three additional ports for asynchronous reads.

In single-port mode, read and write addresses share the same address bus. In dual-port mode, one function generator is connected with the shared read and write port address. The second function generator has the A inputs connected to a second read-only port address and the WA inputs shared with the first read/write port address.

Figure 6 through Figure 14 illustrate various example distributed RAM configurations occupying one SLICEM. When using x2 configuration (RAM32X2Q), A6 and WA6 are driven High by the software to keep O5 and O6 independent.

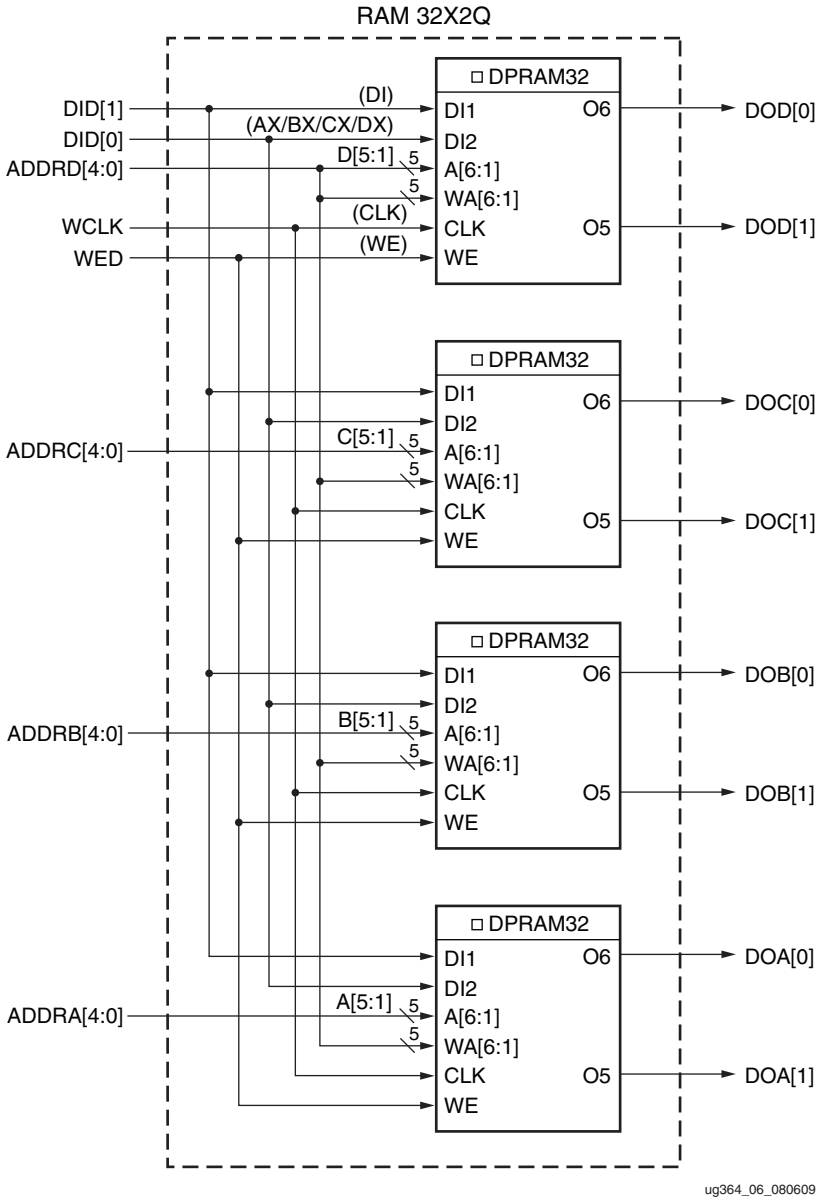

Figure 6: **Distributed RAM (RAM32X2Q)**

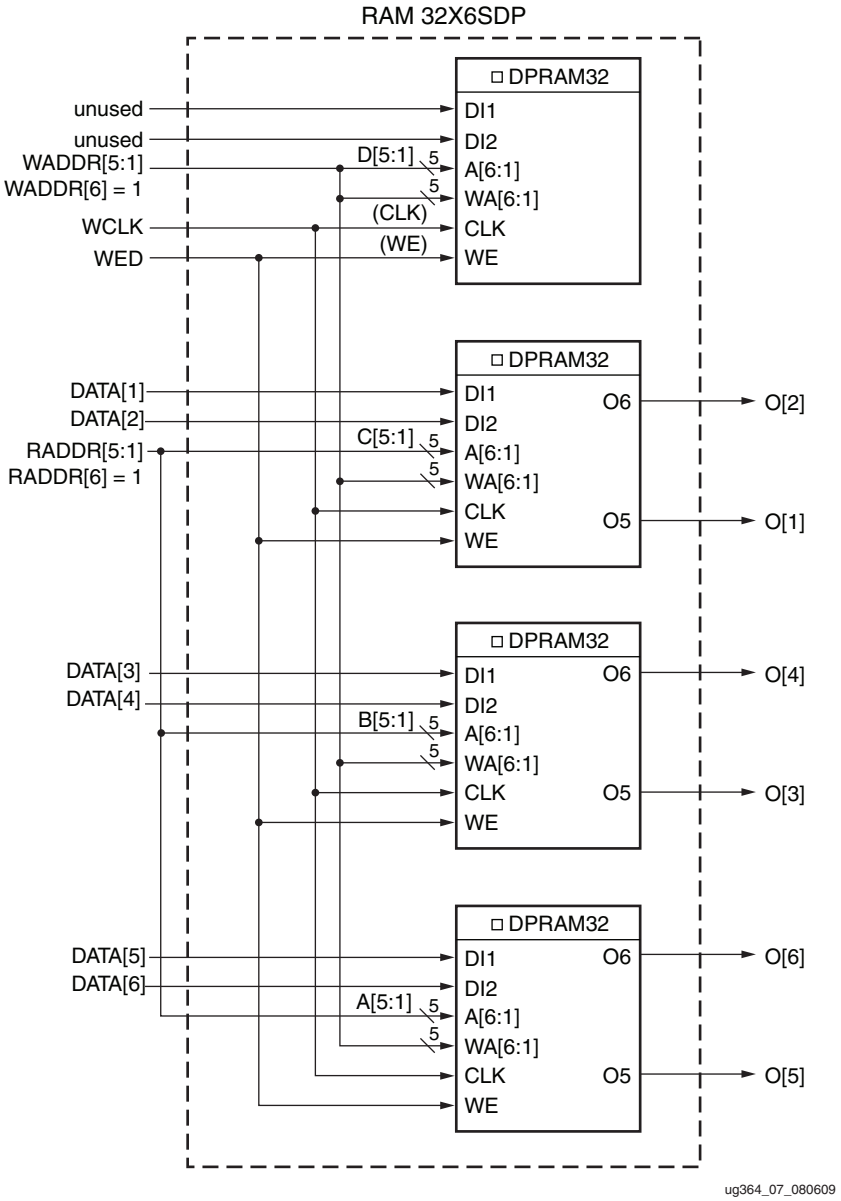

Figure 7: **Distributed RAM (RAM32X6SDP)**

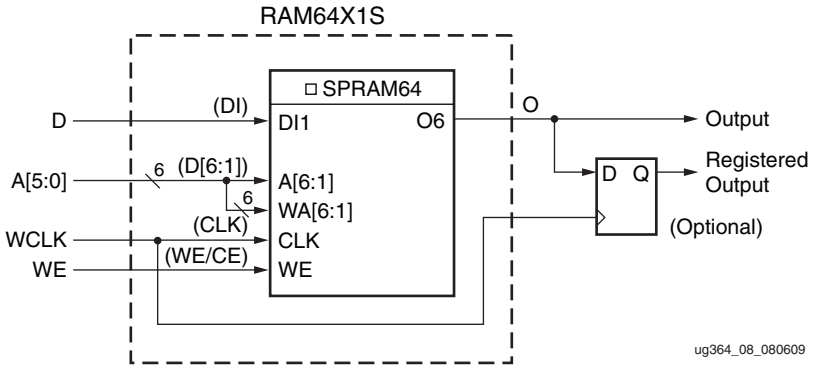

Figure 8: **Distributed RAM (RAM64X1S)**

If four single-port 64 x 1-bit modules are built, the four RAM64X1S primitives can occupy a SLICEM, as long as they share the same clock, write enable, and shared read and write port address inputs. This configuration equates to 64 x 4-bit single-port distributed RAM.

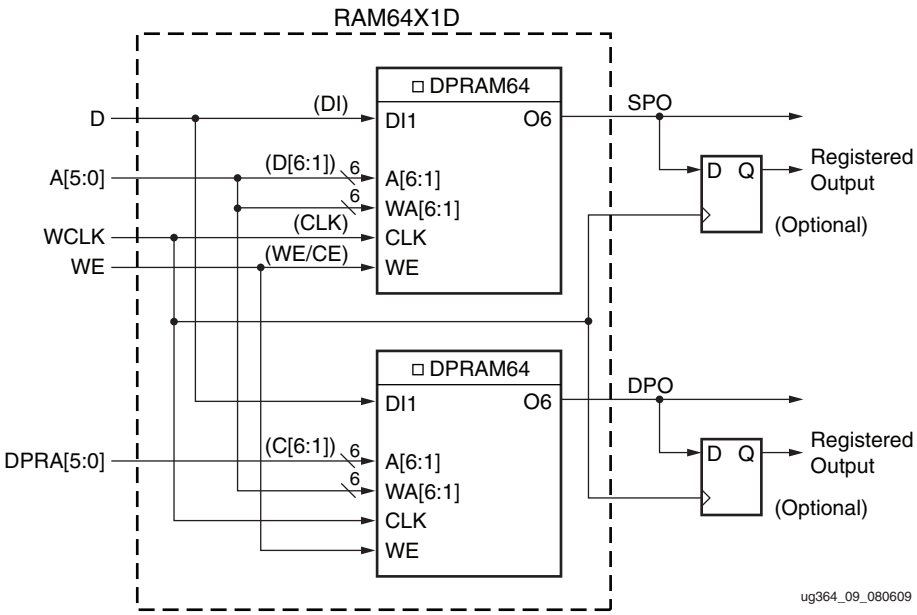

Figure 9: **Distributed RAM (RAM64X1D)**

If two dual-port 64 x 1-bit modules are built, the two RAM64X1D primitives can occupy a SLICEM, as long as they share the same clock, write enable, and shared read and write port address inputs. This configuration equates to 64 x 2-bit dual-port distributed RAM.

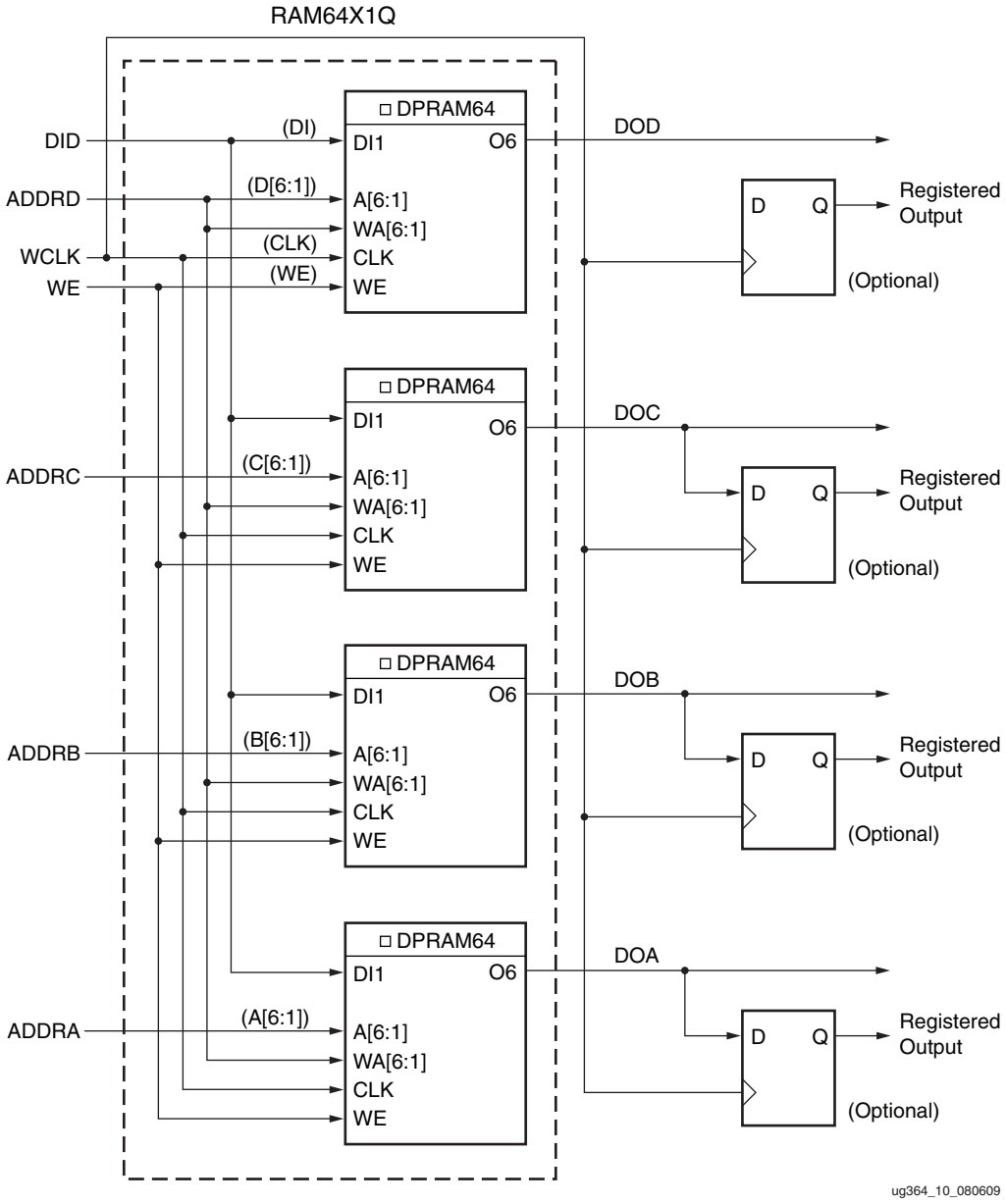

Figure 10: **Distributed RAM (RAM64X1Q)**

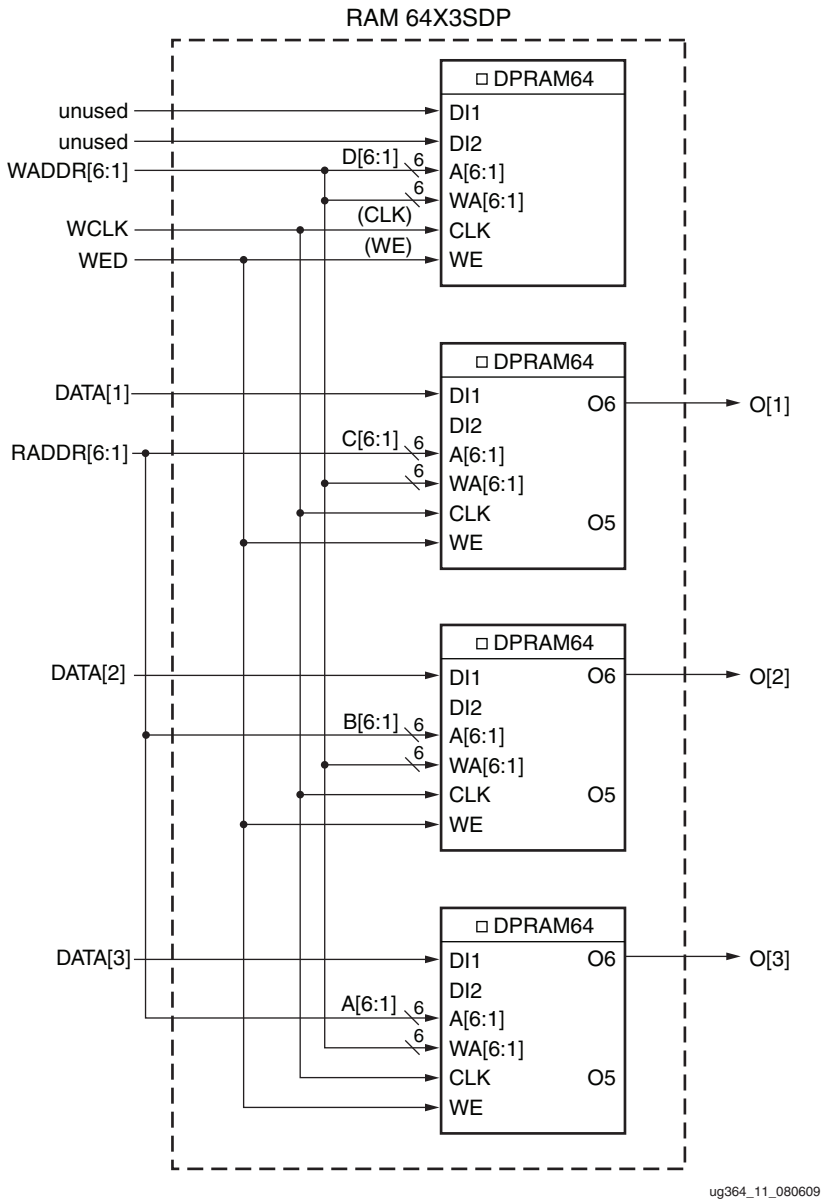

Figure 11: **Distributed RAM (RAM64X3SDP)**

Implementation of distributed RAM configurations with depth greater than 64 requires the usage of wide-function multiplexers (F7AMUX, F7BMUX, and F8MUX).

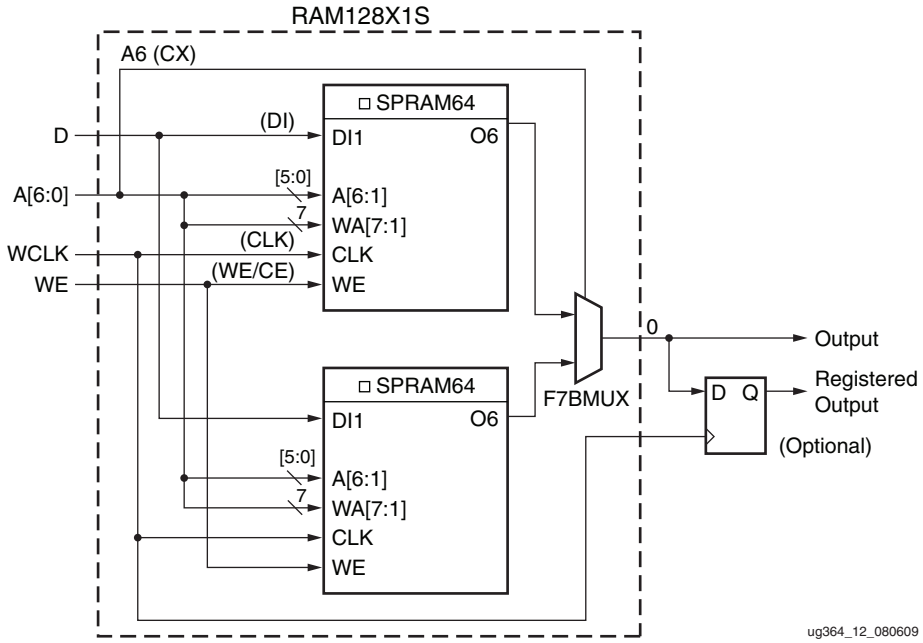

Figure 12: **Distributed RAM (RAM128X1S)**

If two single-port 128 x 1-bit modules are built, the two RAM128X1S primitives can occupy a SLICEM, as long as they share the same clock, write enable, and shared read and write port address inputs. This configuration equates to 128 x 2-bit single-port distributed RAM.

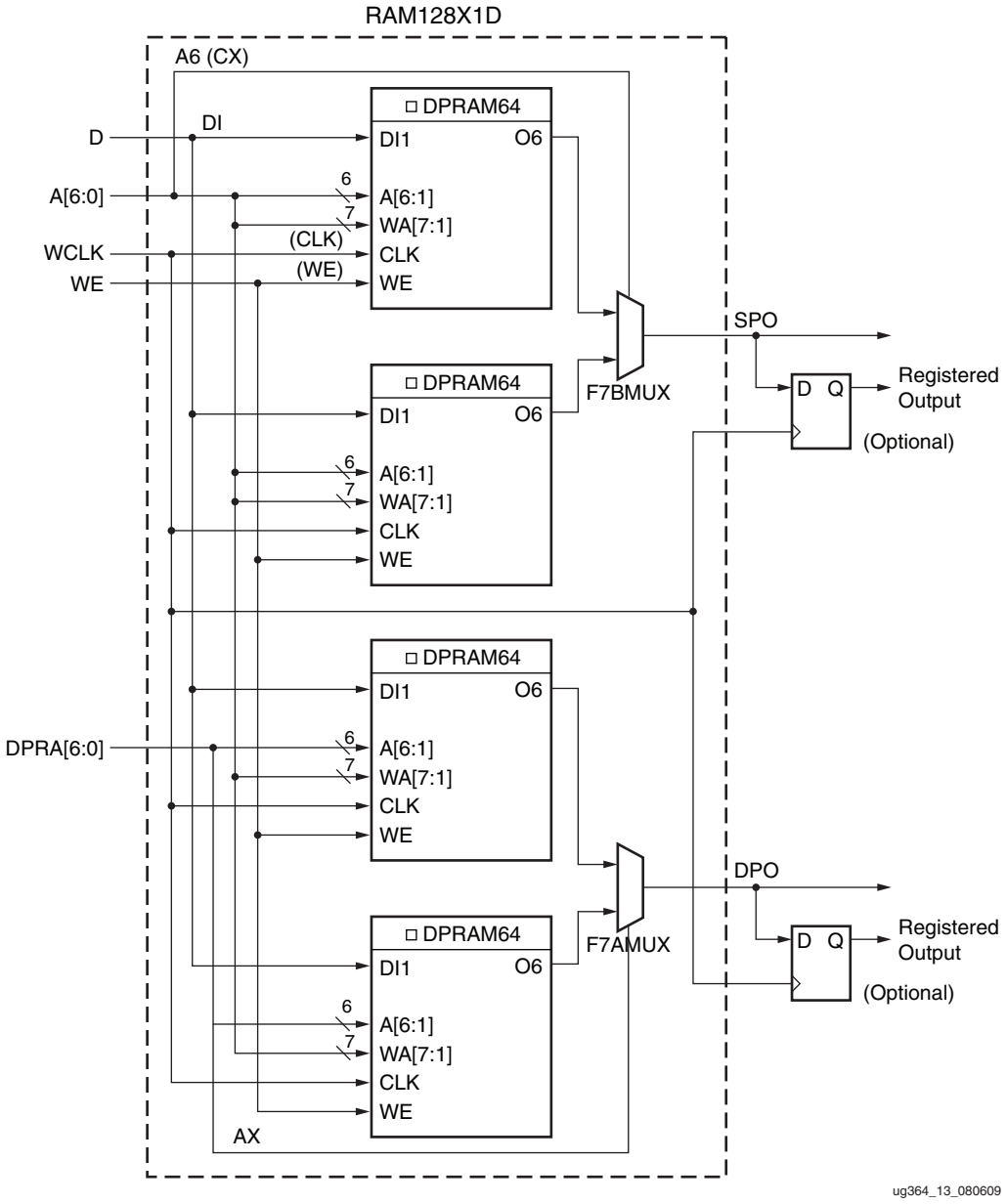

Figure 13: **Distributed RAM (RAM128X1D)**

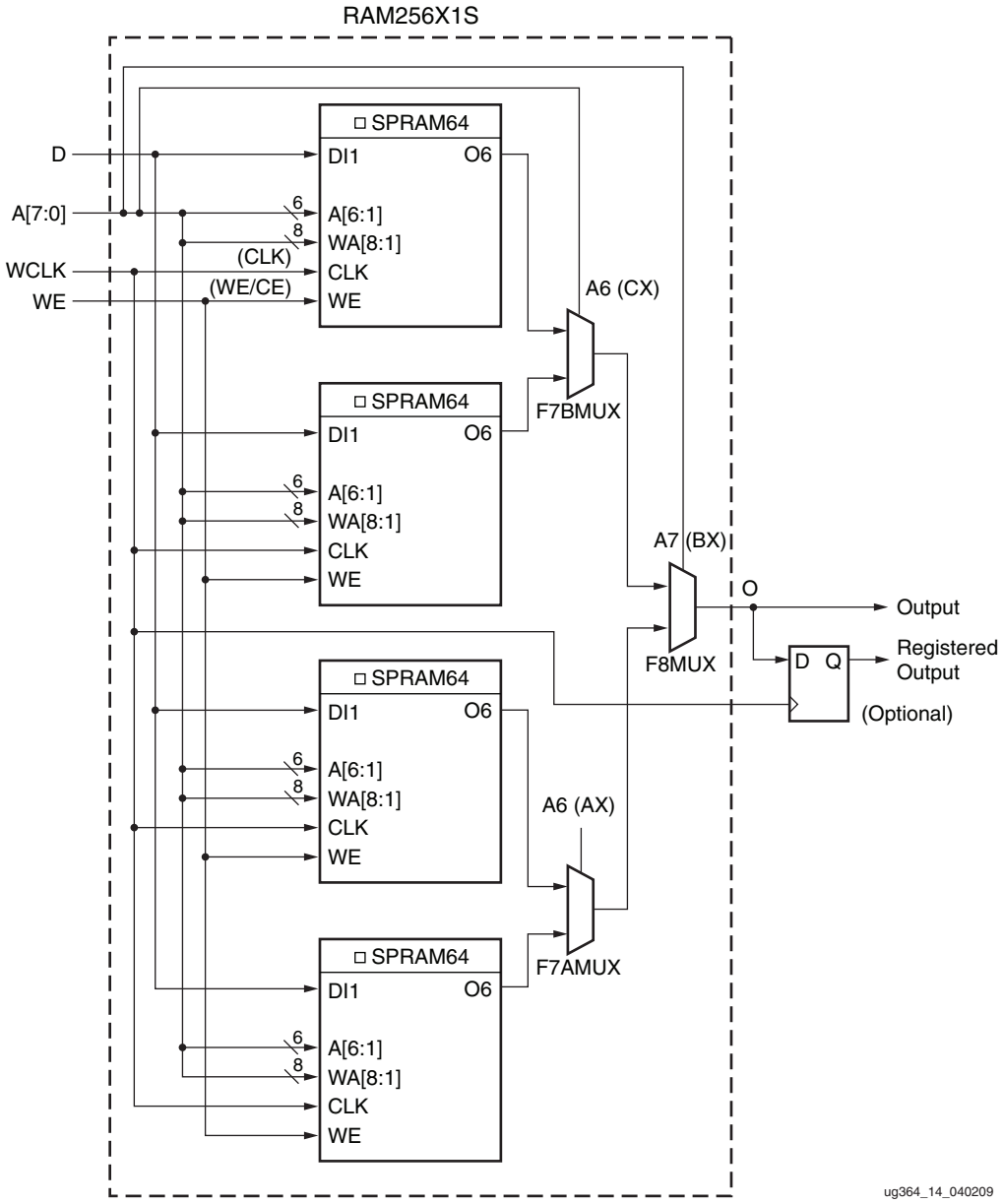

Figure 14: **Distributed RAM (RAM256X1S)**

Distributed RAM configurations greater than the provided examples require more than one SLICEM. There are no direct connections between slices to form larger distributed RAM configurations within a CLB or between slices.

Distributed RAM Data Flow

Synchronous Write Operation

The synchronous write operation is a single clock-edge operation with an active-High write-enable (WE) feature. When WE is High, the input (D) is loaded into the memory location at address A.

Asynchronous Read Operation

The output is determined by the address A (for single-port mode output/SPO output of dual-port mode), or address DPRA (DPO output of dual-port mode). Each time a new address is applied to the address pins, the data value in the memory location of that address is available on the output after the time delay to access the LUT. This operation is asynchronous and independent of the clock signal.

Distributed RAM Summary

- Single-port and dual-port modes are available in SLICEMs.
- A write operation requires one clock edge.
- Read operations are asynchronous (Q output).
- The data input has a setup-to-clock timing specification.

Read Only Memory (ROM)

Each function generator in SLICEMs and SLICELs can implement a 64 x 1-bit ROM. Three configurations are available: ROM64x1, ROM128x1, and ROM256x1. ROM contents are loaded at each device configuration. Table 5 shows the number of LUTs occupied by each ROM configuration.

Table 5: **ROM Configuration**

ROM	Number of LUTs
64 x 1	1
128 x 1	2
256 x 1	4

Shift Registers (Available in SLICEM only)

A SLICEM function generator can also be configured as a 32-bit shift register without using the flip-flops available in a slice. Used in this way, each LUT can delay serial data anywhere from one to 32 clock cycles. The shiftin D (DI1 LUT pin) and shiftout Q31 (MC31 LUT pin) lines cascade LUTs to form larger shift registers. The four LUTs in a SLICEM are thus cascaded to produce delays up to 128 clock cycles. It is also possible to combine shift registers across more than one SLICEM. Note that there are no direct connections between slices to form longer shift registers, nor is the MC31 output at LUT B/C/D available. The resulting programmable delays can be used to balance the timing of data pipelines.

Applications requiring delay or latency compensation use these shift registers to develop efficient designs. Shift registers are also useful in synchronous FIFO and content addressable memory (CAM) designs.

The write operation is synchronous with a clock input (CLK) and an optional clock enable (CE). A dynamic read access is performed through the 5-bit address bus, A[4:0]. The LSB of the LUT is unused and the software automatically ties it to a logic High. The configurable shift registers cannot be set or reset. The read is asynchronous; however, a storage element or flip-flop is available to implement a synchronous read. In this case, the clock-to-out of the flip-flop determines the overall delay and improves performance. However, one additional cycle of clock latency is added. Any of the 32 bits can be read out asynchronously (at the O6 LUT outputs) by varying the 5-bit address. This capability is useful in creating smaller shift registers (less than 32 bits). For example, when building a 13-bit shift register, simply set the address to the 13^{th} bit. Figure 15 is a logic block diagram of a 32-bit shift register.

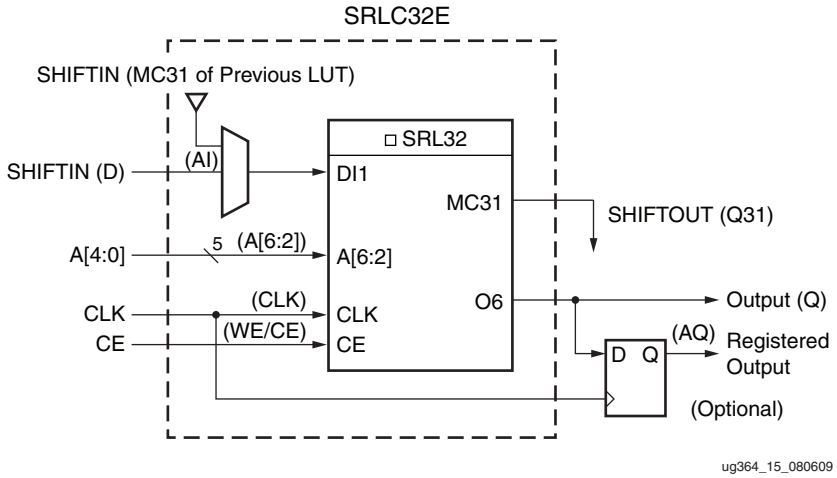

Figure 15: **32-bit Shift Register Configuration**

Figure 16 illustrates an example shift register configuration occupying one function generator.

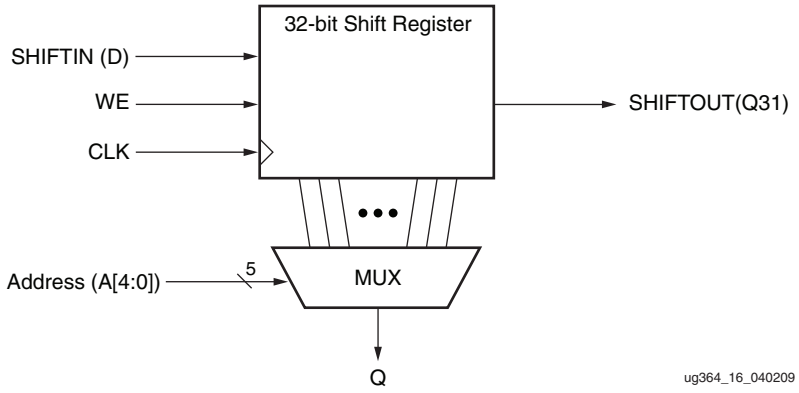

Figure 16: **Representation of a Shift Register**

Figure 17 shows two 16-bit shift registers. The example shown can be implemented in a single LUT.

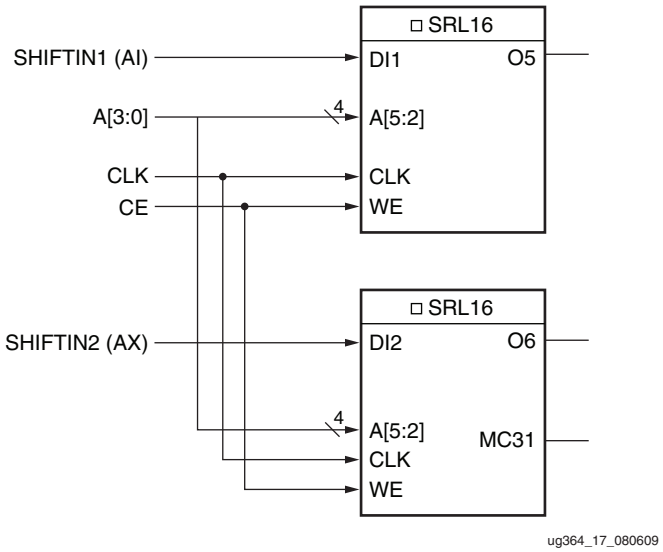

Figure 17: **Dual 16-bit Shift Register Configuration**

As mentioned earlier, an additional output (MC31) and a dedicated connection between shift registers allows connecting the last bit of one shift register to the first bit of the next, without using the LUT O6 output. Longer shift registers can be built with dynamic access to any bit in the chain. The shift register chaining and the F7AMUX, F7BMUX, and F8MUX multiplexers allow up to a 128-bit shift register with addressable access to be implemented in one SLICEM. Figure 18 through Figure 20 illustrate various example shift register configurations that can occupy one SLICEM.

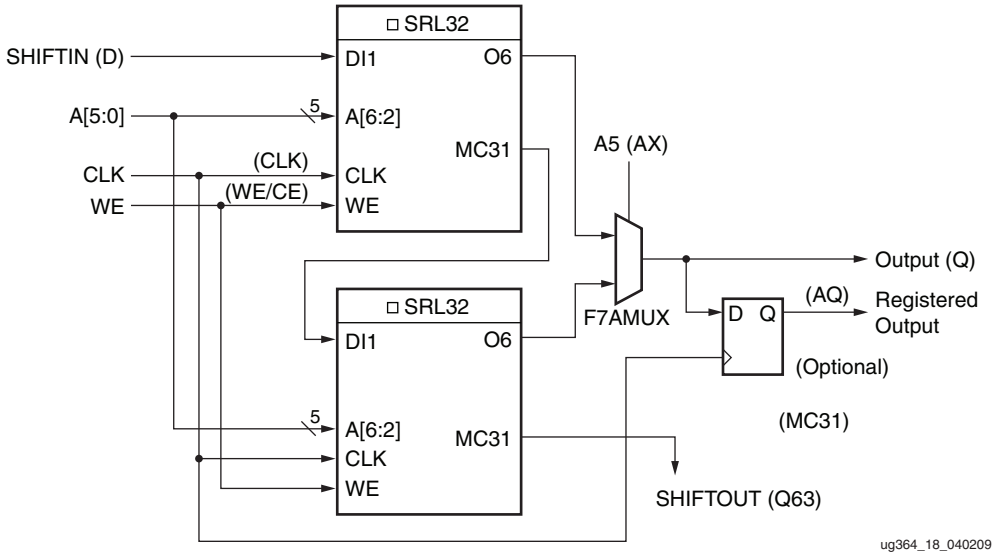

Figure 18: **64-bit Shift Register Configuration**

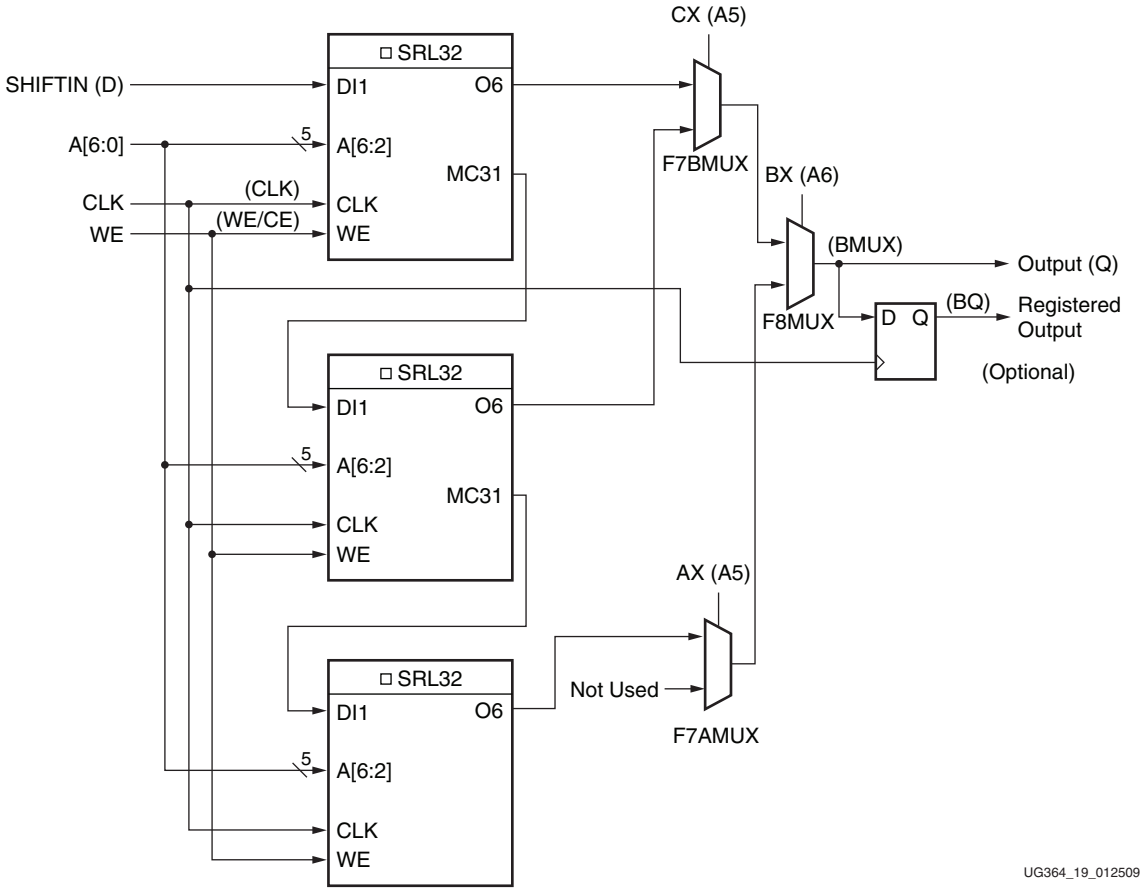

Figure 19: **96-bit Shift Register Configuration**

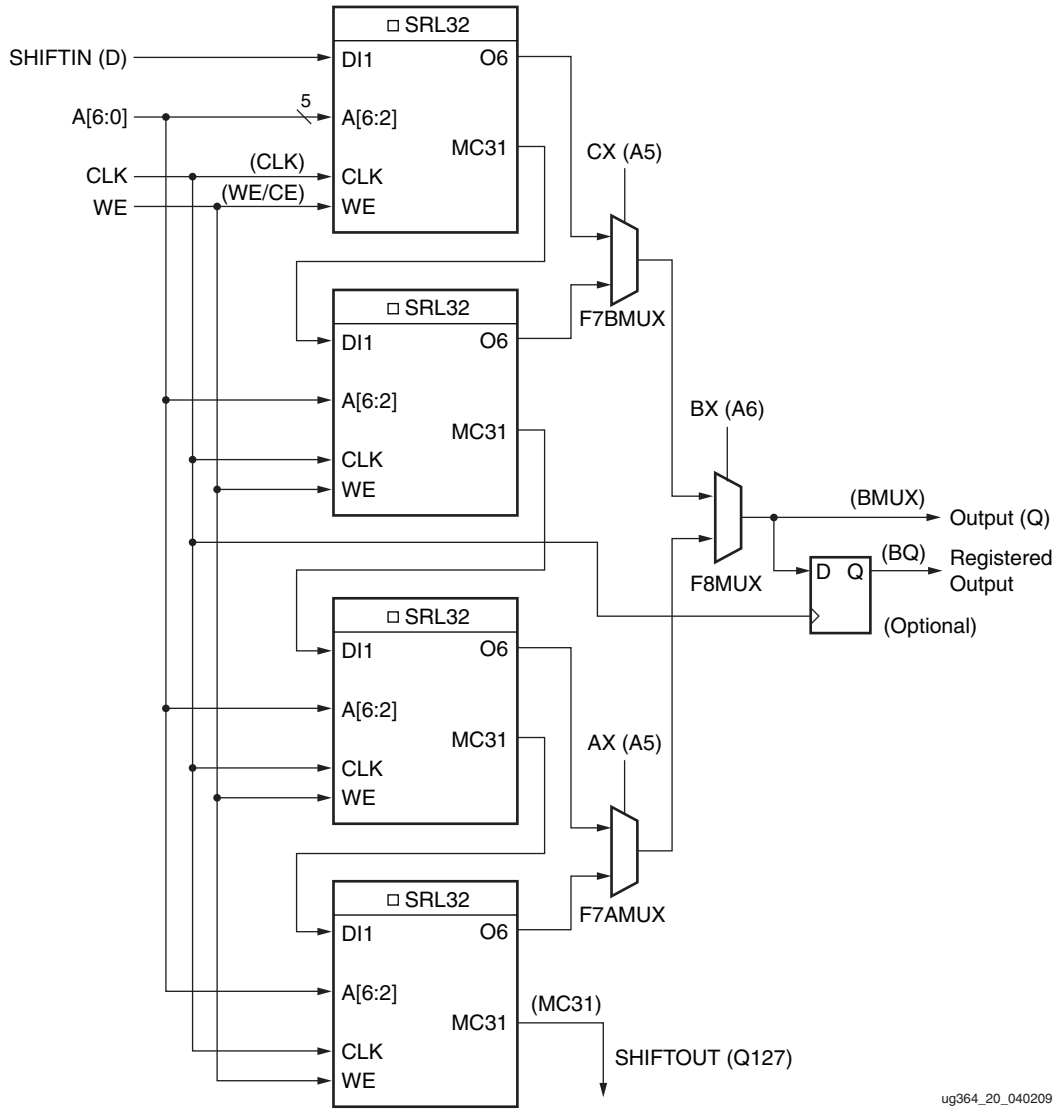

Figure 20: **128-bit Shift Register Configuration**

It is possible to create shift registers longer than 128 bits across more than one SLICEM. However, there are no direct connections between slices to form these shift registers.

Shift Register Data Flow

Shift Operation

The shift operation is a single clock-edge operation, with an active-High clock enable feature. When enable is High, the input (D) is loaded into the first bit of the shift register. Each bit is also shifted to the next highest bit position. In a cascadable shift register configuration, the last bit is shifted out on the M31 output.

The bit selected by the 5-bit address port (A[4:0]) appears on the Q output.

Dynamic Read Operation

The Q output is determined by the 5-bit address. Each time a new address is applied to the 5-input address pins, the new bit position value is available on the Q output after the time

delay to access the LUT. This operation is asynchronous and independent of the clock and clock-enable signals.

Static Read Operation

If the 5-bit address is fixed, the Q output always uses the same bit position. This mode implements any shift-register length from 1 to 32 bits in one LUT. The shift register length is (N+1), where N is the input address (0 – 31).

The Q output changes synchronously with each shift operation. The previous bit is shifted to the next position and appears on the Q output.

Shift Register Summary

- A shift operation requires one clock edge.
- Dynamic-length read operations are asynchronous (Q output).
- Static-length read operations are synchronous (Q output).
- The data input has a setup-to-clock timing specification.
- In a cascadable configuration, the Q31 output always contains the last bit value.
- The Q31 output changes synchronously after each shift operation.

Multiplexers

Function generators and associated multiplexers in Virtex-6 FPGAs can implement the following:

- 4:1 multiplexers using one LUT
- 8:1 multiplexers using two LUTs
- 16:1 multiplexers using four LUTs

These wide input multiplexers are implemented in one level or logic (or LUT) using the dedicated F7AMUX, F7BMUX, and F8MUX multiplexers. These multiplexers allow LUT combinations of up to four LUTs in a slice.

Designing Large Multiplexers

4:1 Multiplexer

Each LUT can be configured into a 4:1 MUX. The 4:1 MUX can be implemented with a flip-flop in the same slice. Up to four 4:1 MUXes can be implemented in a slice, as shown in Figure 21.

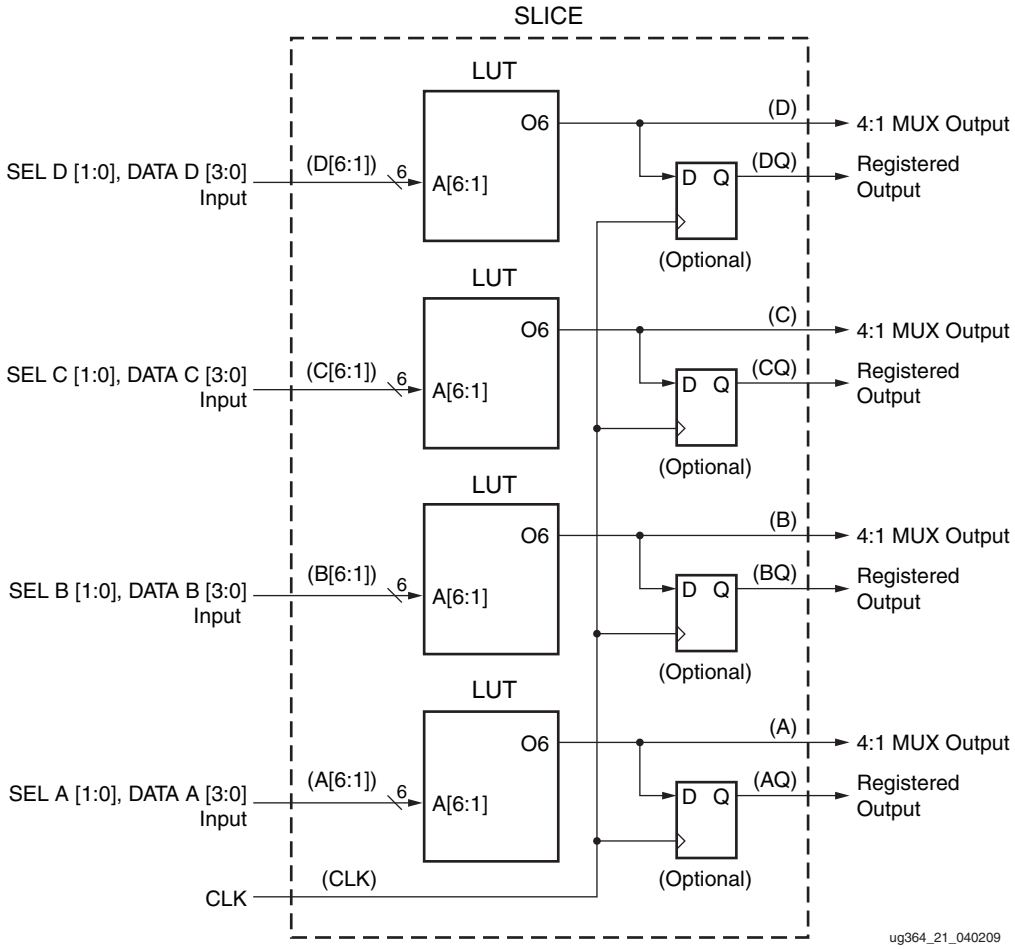

Figure 21: **Four 4:1 Multiplexers in a Slice**

8:1 Multiplexer

Each slice has an F7AMUX and an F7BMUX. These two muxes combine the output of two LUTs to form a combinatorial function up to 13 inputs (or an 8:1 MUX). Up to two 8:1 MUXes can be implemented in a slice, as shown in Figure 22.

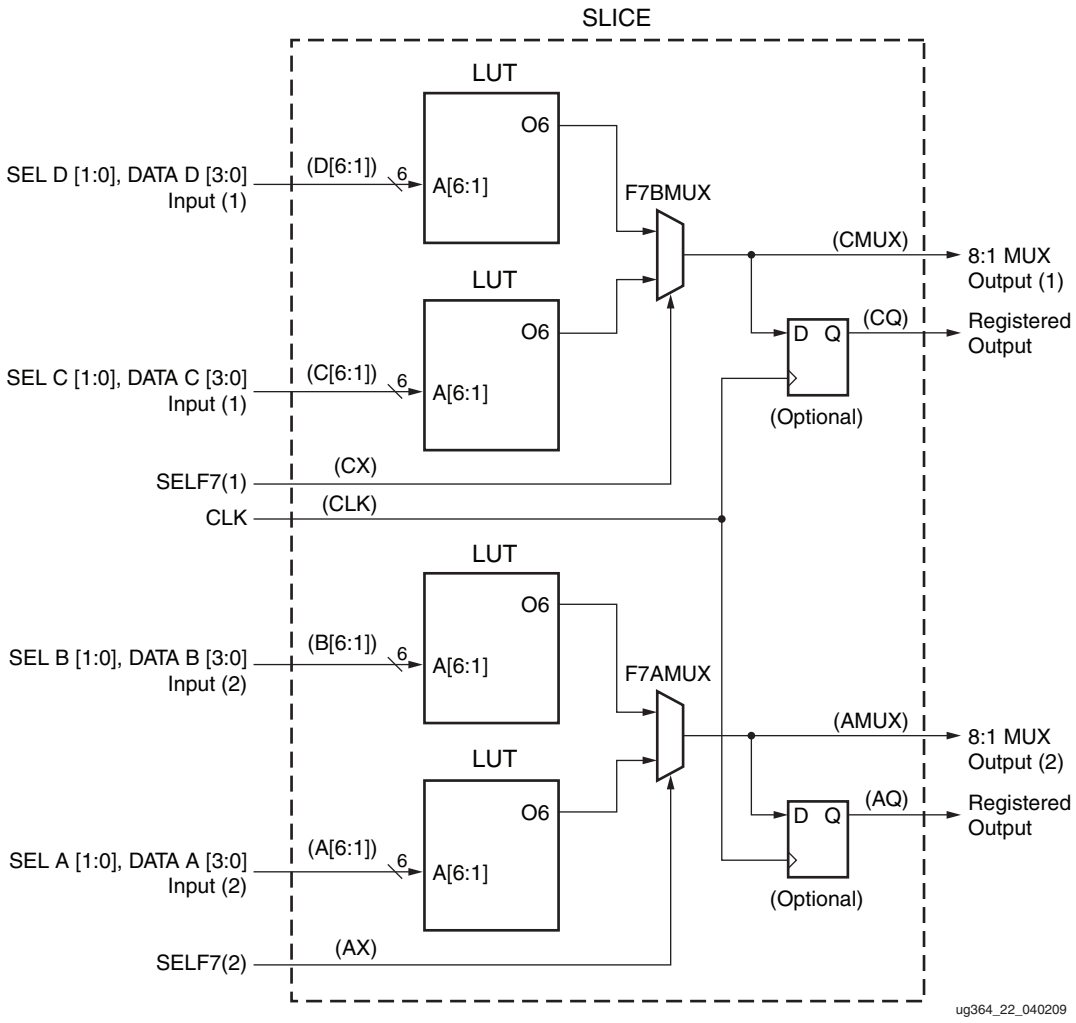

Figure 22: **Two 8:1 Multiplexers in a Slice**

16:1 Multiplexer

Each slice has an F8MUX. F8MUX combines the outputs of F7AMUX and F7BMUX to form a combinatorial function up to 27 inputs (or a 16:1 MUX). Only one 16:1 MUX can be implemented in a slice, as shown in Figure 23.

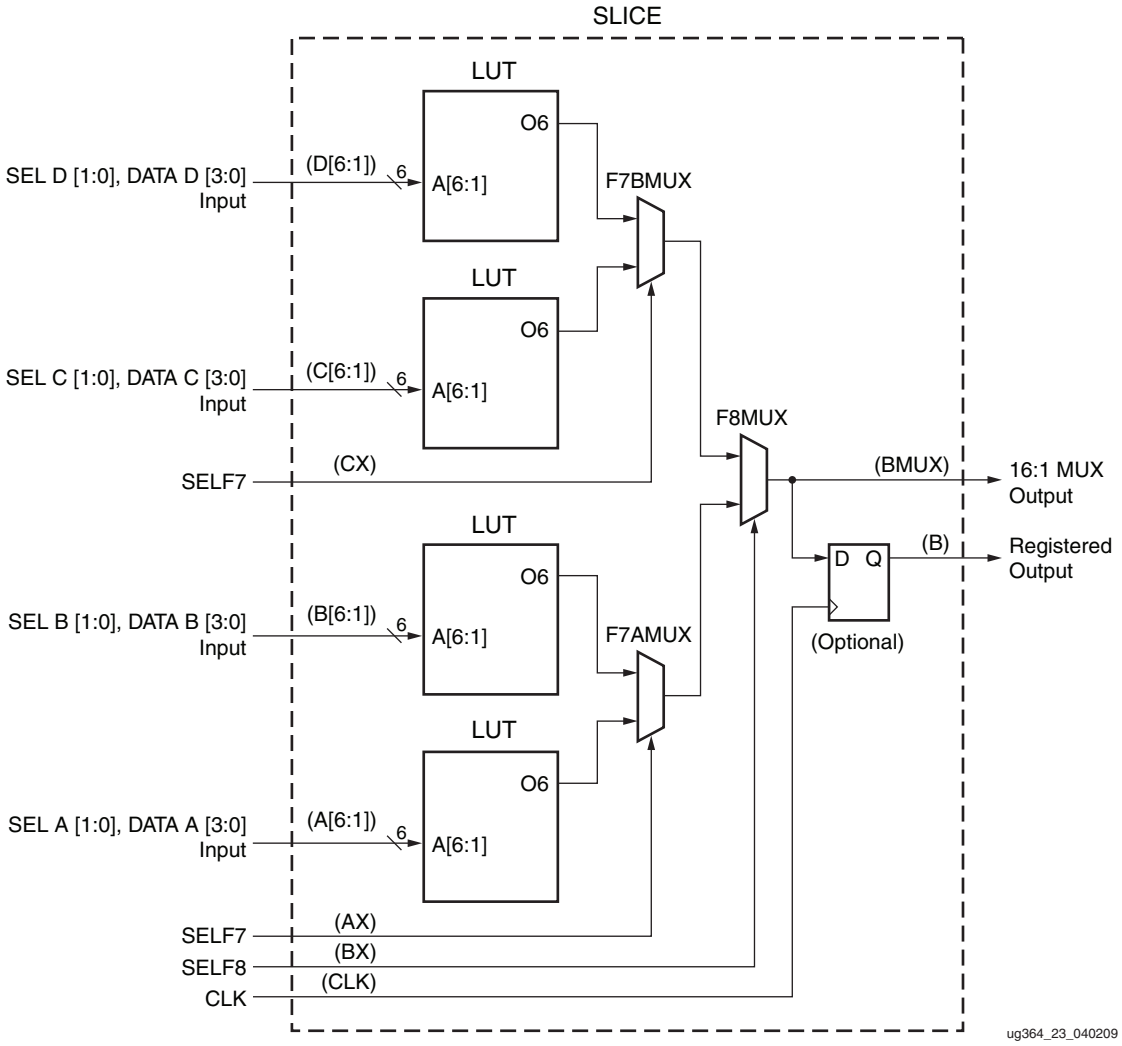

Figure 23: **16:1 Multiplexer in a Slice**

It is possible to create multiplexers wider than 16:1 across more than one SLICEM. However, there are no direct connections between slices to form these wide multiplexers.

Fast Lookahead Carry Logic

In addition to function generators, dedicated carry logic is provided to perform fast arithmetic addition and subtraction in a slice. A Virtex-6 FPGA CLB has two separate carry chains, as shown in Figure 1. The carry chains are cascadable to form wider add/subtract logic, as shown in Figure 2.

The carry chain in the Virtex-6 device is running upward and has a height of four bits per slice. For each bit, there is a carry multiplexer (MUXCY) and a dedicated XOR gate for adding/subtracting the operands with a selected carry bits. The dedicated carry path and carry multiplexer (MUXCY) can also be used to cascade function generators for implementing wide logic functions.

Figure 24 illustrates the carry chain with associated logic elements in a slice.

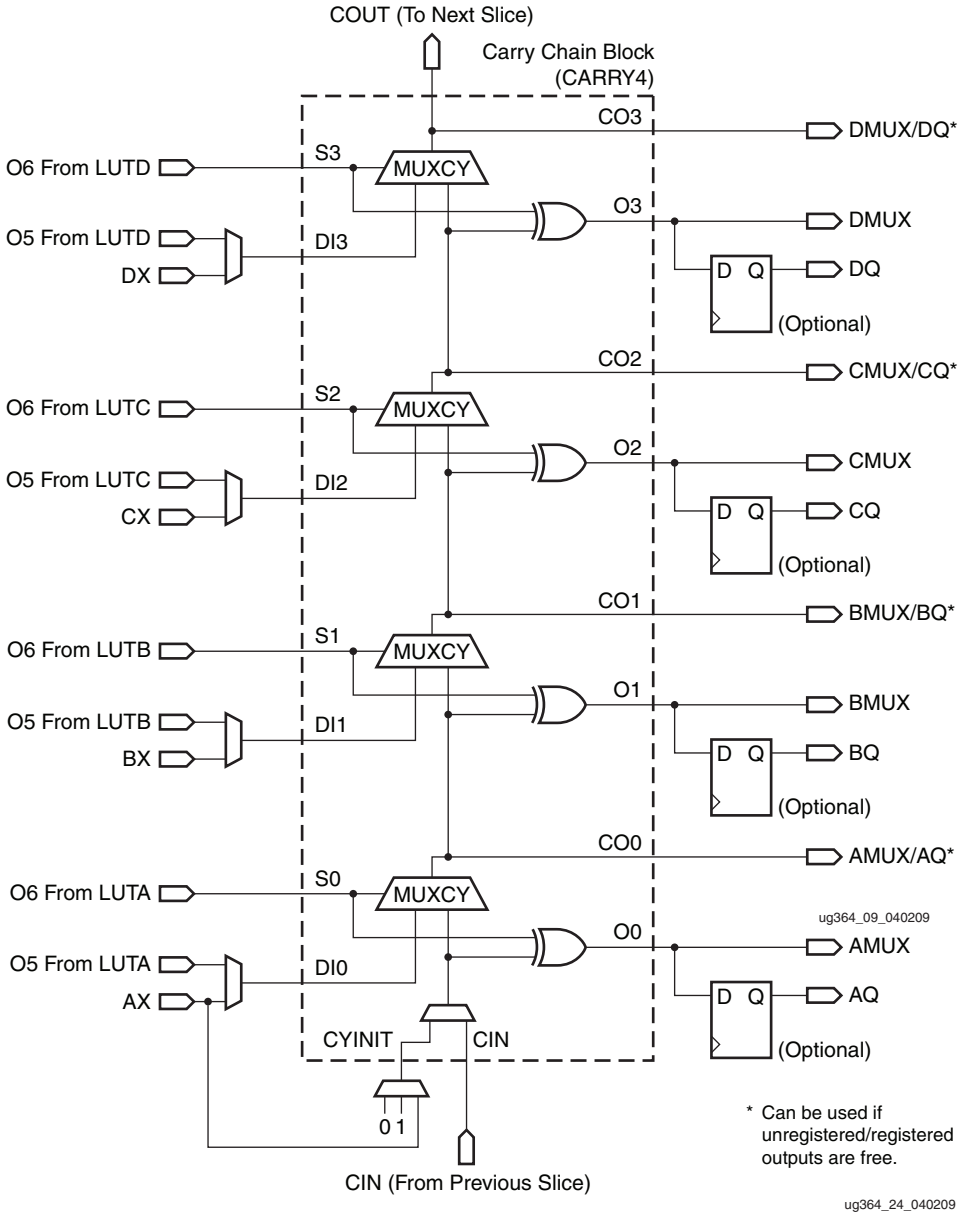

Figure 24: **Fast Carry Logic Path and Associated Elements**

The carry chains carry lookahead logic along with the function generators. There are ten independent inputs (S inputs – S0 to S3, DI inputs – DI1 to DI4, CYINIT and CIN) and eight independent outputs (O outputs – O0 to O3, and CO outputs – CO0 to CO3).

The S inputs are used for the "propagate" signals of the carry lookahead logic. The "propagate" signals are sourced from the O6 output of a function generator. The DI inputs are used for the "generate" signals of the carry lookahead logic. The "generate" signals are sourced from either the O5 output of a function generator or the BYPASS input (AX, BX, CX, or DX) of a slice. The former input is used to create a multiplier, while the latter is used to create an adder/accumulator. CYINIT is the CIN of the first bit in a carry chain. The CYINIT value can be 0 (for add), 1 (for subtract), or AX input (for the dynamic first carry bit). The CIN input is used to cascade slices to form a longer carry chain. The O outputs contain the sum of the addition/subtraction. The CO outputs compute the carry out for

each bit. CO3 is connected to COUT output of a slice to form a longer carry chain by cascading multiple slices. The propagation delay for an adder increases linearly with the number of bits in the operand, as more carry chains are cascaded. The carry chain can be implemented with a storage element or a flip-flop in the same slice.

CLB / Slice Timing Models

Due to the large size and complexity of Virtex-6 FPGAs, understanding the timing associated with the various paths and functional elements is a difficult and important task. Although it is not necessary to understand the various timing parameters to implement most designs using Xilinx software, a thorough timing model can assist advanced users in analyzing critical paths or planning speed-sensitive designs.

Three timing model sections are described:

- Functional element diagram – basic architectural schematic illustrating pins and connections
- Timing parameters – definitions of *Virtex-6 FPGA Data Sheet* timing parameters
- Timing Diagram - illustrates functional element timing parameters relative to each other

Use the models in this chapter in conjunction with both the Xilinx Timing Analyzer software (TRCE) and the section on switching characteristics in the *Virtex-6 FPGA Data Sheet*. All pin names, parameter names, and paths are consistent with the post-route timing and pre-route static timing reports. Most of the timing parameters found in the section on switching characteristics are described in this chapter.

All timing parameters reported in the *Virtex-6 FPGA Data Sheet* are associated with slices and CLBs. The following sections correspond to specific switching characteristics sections in the *Virtex-6 FPGA Data Sheet*:

- General Slice Timing Model and Parameters (CLB Switching Characteristics)
- Slice Distributed RAM Timing Model and Parameters (Available in SLICEM only) (CLB Distributed RAM Switching Characteristics)
- Slice SRL Timing Model and Parameters (Available in SLICEM only) (CLB SRL Switching Characteristics)
- Slice Carry-Chain Timing Model and Parameters (CLB Application Switching Characteristics)

General Slice Timing Model and Parameters

A simplified Virtex-6 FPGA slice is shown in Figure 25. Some elements of the slice are omitted for clarity. Only the elements relevant to the timing paths described in this section are shown.

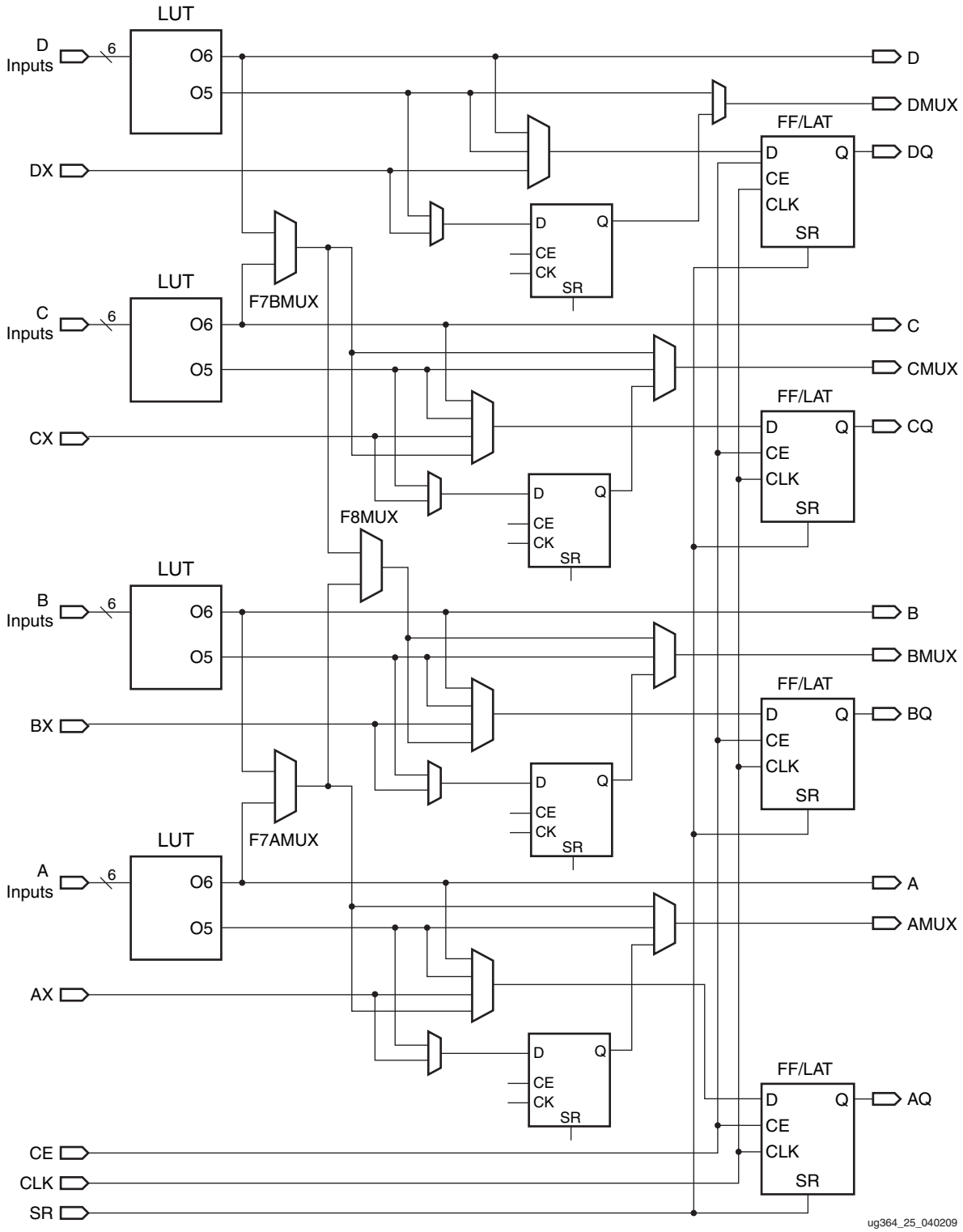

Figure 25: **Simplified Virtex-6 FPGA Slice**

Timing Parameters

Table 6 shows the general slice timing parameters for a majority of the paths in Figure 25.

Table 6: **General Slice Timing Parameters**

Parameter	Function	Description
Combinatorial Delays		
T_{ILO}(1)	A/B/C/D inputs to A/B/C/D outputs	Propagation delay from the A/B/C/D inputs of the slice, through the look-up tables (LUTs), to the A/B/C/D outputs of the slice (six-input function).
T_{ILO_2}	A/B/C/D inputs to AMUX/CMUX outputs	Propagation delay from the A/B/C/D inputs of the slice, through the LUTs and F7AMUX/F7BMUX to the AMUX/CMUX outputs (seven-input function).
T_{ILO_3}	A/B/C/D inputs to BMUX output	Propagation delay from the A/B/C/D inputs of the slice, through the LUTs, F7AMUX/F7BMUX, and F8MUX to the BMUX output (eight-input function).
Sequential Delays		
T_{CKO} Flip-Flop/ Latch element	FF Clock (CLK) to AQ/BQ/CQ/DQ outputs	Time after the clock that data is stable at the AQ/BQ/CQ/DQ outputs of the slice sequential elements (configured as a flip-flop).
T_{CKO} Flip-Flop only element	FF Clock (CLK) to AQ/BQ/CQ/DQ outputs	Time after the clock that data is stable at the AQ/BQ/CQ/DQ outputs of the slice sequential elements.
T_{CKLO}	Latch Clock (CLK) to AQ/BQ/CQ/DQ outputs	Time after the clock that data is stable at the AQ/BQ/CQ/DQ outputs of the slice sequential elements (configured as a latch).
Setup and Hold Times for Slice Sequential Elements(2)		
T_{DICK}/T_{CKDI} Flip-Flop/ Latch element	AX/BX/CX/DX inputs	Time before/after the CLK that data from the AX/BX/CX/DX inputs of the slice must be stable at the D input of the slice sequential elements (configured as a flip-flop).
T_{DICK}/T_{CKDI} Flip-Flop only element	AX/BX/CX/DX inputs	Time before/after the CLK that data from the AX/BX/CX/DX inputs of the slice must be stable at the D input of the slice sequential elements.
T_{CECK}/T_{CKCE} Flip-Flop/ Latch element	CE input	Time before/after the CLK that the CE input of the slice must be stable at the CE input of the slice sequential elements (configured as a flip-flop).
T_{CECK}/T_{CKCE} Flip-Flop only element	CE input	Time before/after the CLK that the CE input of the slice must be stable at the CE input of the slice sequential elements.
T_{SRCK}/T_{CKSR} Flip-Flop/ Latch element	SR input	Time before/after the CLK that the SR (Set/Reset) of the slice must be stable at the SR inputs of the slice sequential elements (configured as a flip-flop).

Table 6: **General Slice Timing Parameters *(Cont'd)***

Parameter	Function	Description
T_{SRCK}/T_{CKSR} Flip-Flop only element	SR input	Time before/after the CLK that the SR (Set/Reset) inputs of the slice must be stable at the SR inputs of the slice sequential elements
Set/Reset		
T_{RPW}		Minimum Pulse Width for the SR (Set/Reset).
T_{RQ}		Propagation delay for an asynchronous Set/Reset of the slice sequential elements. From the SR inputs to the AQ/BQ/CQ/DQ outputs.
F_{TOG}		Toggle Frequency – Maximum frequency that a CLB flip-flop can be clocked: $1/(T_{CH} + T_{CL})$.

Notes:

1. This parameter includes a LUT configured as two five-input functions.
2. T_{XXCK} = Setup Time (before clock edge), and T_{CKXX} = Hold Time (after clock edge).

Timing Characteristics

Figure 26 illustrates the general timing characteristics of a Virtex-6 FPGA slice.

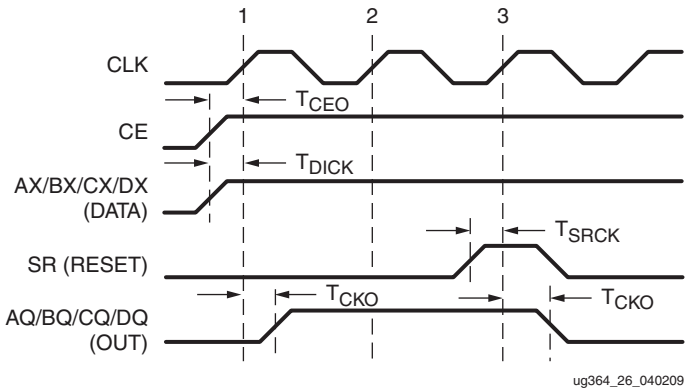

Figure 26: **General Slice Timing Characteristics**

- At time T_{CEO} before clock event (1), the clock-enable signal becomes valid-high at the CE input of the slice register.
- At time T_{DICK} before clock event (1), data from either AX, BX, CX, or DX inputs become valid-high at the D input of the slice register and is reflected on either the AQ, BQ, CQ, or DQ pin at time T_{CKO} after clock event (1).
- At time T_{SRCK} before clock event (3), the SR signal (configured as synchronous reset) becomes valid-high, resetting the slice register. This is reflected on the AQ, BQ, CQ, or DQ pin at time T_{CKO} after clock event (3).

Slice Distributed RAM Timing Model and Parameters (Available in SLICEM only)

Figure 27 illustrates the details of distributed RAM implemented in a Virtex-6 FPGA slice. Some elements of the slice are omitted for clarity. Only the elements relevant to the timing paths described in this section are shown.

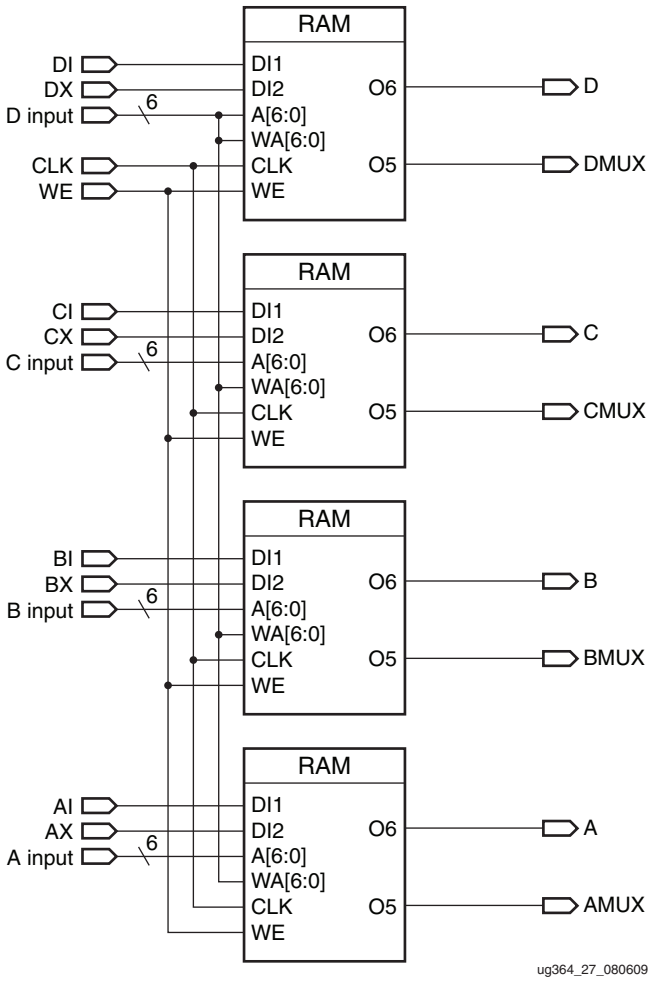

Figure 27: **Simplified Virtex-6 FPGA SLICEM Distributed RAM**

Distributed RAM Timing Parameters

Table 7 shows the timing parameters for the distributed RAM in SLICEM for a majority of the paths in Figure 27.

Table 7: **Distributed RAM Timing Parameters**

Parameter	Function	Description
Sequential Delays for a Slice LUT Configured as RAM (Distributed RAM)		
T_{SHCKO}(1)	CLK to A/B/C/D outputs	Time after the CLK of a write operation that the data written to the distributed RAM is stable on the A/B/C/D output of the slice.
Setup and Hold Times for a Slice LUT Configured as RAM (Distributed RAM)(2)		
T_{DS}/T_{DH}(3)	AI/BI/CI/DI configured as data input (DI1)	Time before/after the clock that data must be stable at the AI/BI/CI/DI input of the slice.
T_{ACK}/T_{CKA}	A/B/C/D address inputs	Time before/after the clock that address signals must be stable at the A/B/C/D inputs of the slice LUT (configured as RAM).
T_{WS}/T_{WH}	WE input	Time before/after the clock that the write enable signal must be stable at the WE input of the slice LUT (configured as RAM).
Clock CLK		
T_{WPH}		Minimum Pulse Width, High
T_{WPL}		Minimum Pulse Width, Low
T_{WC}		Minimum clock period to meet address write cycle time.

Notes:

1. This parameters includes a LUT configured as a two-bit distributed RAM.
2. T_{XXCK} = Setup Time (before clock edge), and T_{CKXX} = Hold Time (after clock edge).
3. Parameter includes AX/BX/CX/DX configured as a data input (DI2).

Distributed RAM Timing Characteristics

The timing characteristics of a 16-bit distributed RAM implemented in a Virtex-6 FPGA slice (LUT configured as RAM) are shown in Figure 28.

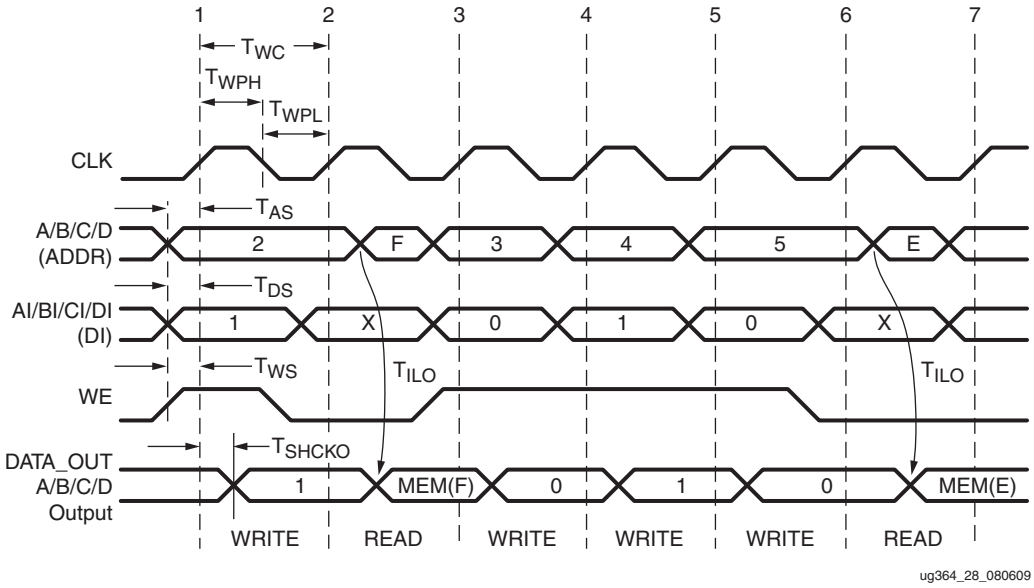

Figure 28: **Slice Distributed RAM Timing Characteristics**

Clock Event 1: Write Operation

During a Write operation, the contents of the memory at the address on the ADDR inputs are changed. The data written to this memory location is reflected on the A/B/C/D outputs synchronously.

- At time T_{WS} before clock event 1, the write-enable signal (WE) becomes valid-high, enabling the RAM for a Write operation.
- At time T_{AS} before clock event 1, the address (2) becomes valid at the A/B/C/D inputs of the RAM.
- At time T_{DS} before clock event 1, the DATA becomes valid (1) at the DI input of the RAM and is reflected on the A/B/C/D output at time T_{SHCKO} after clock event 1.

This is also applicable to the AMUX, BMUX, CMUX, DMUX, and COUT outputs at time T_{SHCKO} and T_{WOSCO} after clock event 1.

Clock Event 2: Read Operation

All Read operations are asynchronous in distributed RAM. As long as WE is Low, the address bus can be asserted at any time. The contents of the RAM on the address bus are reflected on the A/B/C/D outputs after a delay of length T_{ILO} (propagation delay through a LUT). The address (F) is asserted *after* clock event 2, and the contents of the RAM at address (F) are reflected at the output after a delay of length T_{ILO}.

Slice SRL Timing Model and Parameters (Available in SLICEM only)

Figure 29 illustrates shift register implementation in a Virtex-6 FPGA slice. Some elements of the slice have been omitted for clarity. Only the elements relevant to the timing paths described in this section are shown.

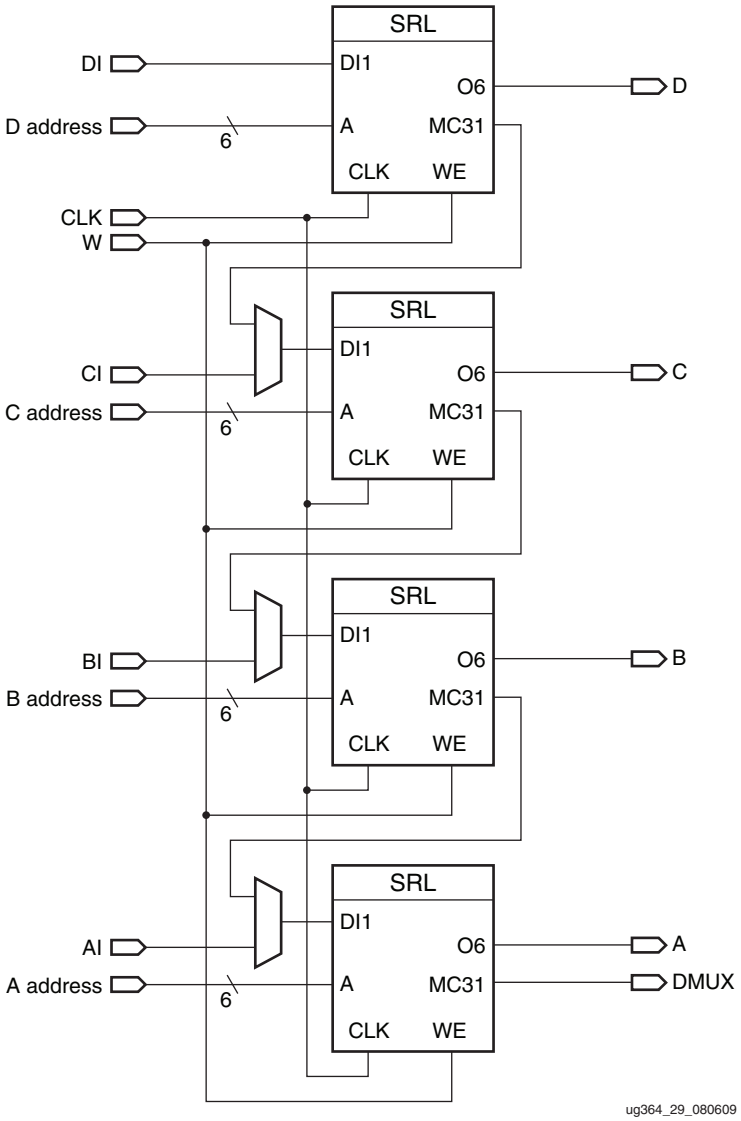

Figure 29: **Simplified Virtex-6 FPGA Slice SRL**

Slice SRL Timing Parameters

Table 8 shows the SLICEM SRL timing parameters for a majority of the paths in Figure 29.

Table 8: **Slice SRL Timing Parameters**

Parameter	Function	Description
Sequential Delays for a Slice LUT Configured as an SRL		
T_{REG}(1)	CLK to A/B/C/D outputs	Time after the CLK of a write operation that the data written to the SRL is stable on the A/B/C/D outputs of the slice.
T_{REG_MUX}(1)	CLK to AMUX - DMUX output	Time after the CLK of a write operation that the data written to the SRL is stable on the DMUX output of the slice.
T_{REG_M31}	CLK to DMUX output via MC31 output	Time after the CLK of a write operation that the data written to the SRL is stable on the DMUX output via MC31 output.
Setup and Hold Times for a Slice LUT Configured SRL(2)		
T_{WS}/T_{WH}	CE input (WE)	Time before/after the clock that the write enable signal must be stable at the WE input of the slice LUT (configured as an SRL).
T_{DS}/T_{DH}(3)	AI/BI/CI/DI configured as data input (DI)	Time before the clock that the data must be stable at the AI/BI/CI/DI input of the slice (configured as an SRL).

Notes:

1. This parameter includes a LUT configured as a two-bit shift register.
2. T_{XXCK} = Setup Time (before clock edge), and T_{CKXX} = Hold Time (after clock edge).
3. Parameter includes AX/BX/CX/DX configured as a data input (DI2) or two bits with a common shift.

Slice SRL Timing Characteristics

Figure 30 illustrates the timing characteristics of a 16-bit shift register implemented in a Virtex-6 FPGA slice (a LUT configured as an SRL).

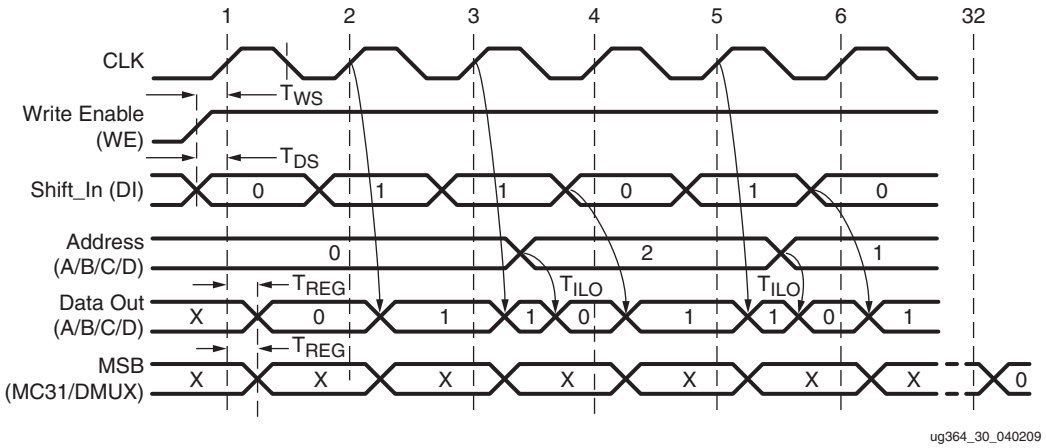

Figure 30: **Slice SRL Timing Characteristics**

Clock Event 1: Shift In

During a write (Shift In) operation, the single-bit content of the register at the address on the A/B/C/D inputs is changed, as data is shifted through the SRL. The data written to this register is reflected on the A/B/C/D outputs synchronously, if the address is unchanged during the clock event. If the A/B/C/D inputs are changed during a clock event, the value of the data at the addressable output (A/B/C/D outputs) is invalid.

- At time T_{WS} before clock event 1, the write-enable signal (WE) becomes valid-High, enabling the SRL for the Write operation that follows.
- At time T_{DS} before clock event 1 the data becomes valid (0) at the DI input of the SRL and is reflected on the A/B/C/D output after a delay of length T_{REG} after clock event 1. Since the address 0 is specified at clock event 1, the data on the DI input is reflected at A/B/C/D output, because it is written to register 0.

Clock Event 2: Shift In

- At time T_{DS} before clock event 2, the data becomes valid (1) at the DI input of the SRL and is reflected on the A/B/C/D output after a delay of length T_{REG} after clock event 2. Since the address 0 is still specified at clock event 2, the data on the DI input is reflected at the D output, because it is written to register 0.

Clock Event 3: Shift In/Addressable (Asynchronous) READ

All Read operations are asynchronous to the CLK signal. If the address is changed (between clock events), the contents of the register at that address are reflected at the addressable output (A/B/C/D outputs) after a delay of length T_{ILO} (propagation delay through a LUT).

- At time T_{DS} before clock event 3, the data becomes valid (1) at the DI input of the SRL and is reflected on the A/B/C/D output T_{REG} time after clock event 3.
- The address is changed (from 0 to 2). The value stored in register 2 at this time is a 0 (in this example, this was the first data shifted in), and it is reflected on the A/B/C/D output after a delay of length T_{ILO}.

Clock Event 32: MSB (Most Significant Bit) Changes

At time T_{REG} after clock event 32, the first bit shifted into the SRL becomes valid (logical 0 in this case) on the DMUX output of the slice via the MC31 output of LUT A (SRL). This is also applicable to the AMUX, BMUX, CMUX, DMUX, and COUT outputs at time T_{REG} and T_{WOSCO} after clock event 1.

Slice Carry-Chain Timing Model and Parameters

Figure 24, page 33 illustrates a carry chain in a Virtex-6 FPGA slice. Some elements of the slice have been omitted for clarity. Only the elements relevant to the timing paths described in this section are shown.

Slice Carry-Chain Timing Parameters

Table 9 shows the slice carry-chain timing parameters for a majority of the paths in Figure 24, page 33.

Table 9: **Slice Carry-Chain Timing Parameters**

Parameter	Function	Description
Sequential Delays for Slice LUT Configured as Carry Chain		
$T_{AXCY}/T_{BXCY}/T_{CXCY}/T_{DXCY}$	AX/BX/CX/DX input to COUT output	Propagation delay from the AX/BX/CX/DX inputs of the slice to the COUT output of the slice.
T_{BYP}	CIN input to COUT output	Propagation delay from the CIN input of the slice to the COUT output of the slice.
$T_{OPCYA}/T_{OPCYB}/T_{OPCYC}/T_{OPCYD}$	A/B/C/D input to COUT output	Propagation delay from the A/B/C/D inputs of the slice to the COUT output of the slice.
$T_{CINA}/T_{CINB}/T_{CINC}/T_{CIND}$	A/B/C/D input to AMUX/BMUX/CMUX/DMUX output	Propagation delay from the A/B/C/D inputs of the slice to AMUX/BMUX/CMUX/DMUX output of the slice using XOR (sum).
Setup and Hold Times for a Slice LUT Configured as a Carry Chain(1)		
T_{CINCK}/T_{CKCIN}	CIN Data inputs	Time before the CLK that data from the CIN input of the slice must be stable at the D input of the slice sequential elements (configured as a flip-flop).

Notes:

1. T_{XXCK} = Setup Time (before clock edge), and T_{CKXX} = Hold Time (after clock edge).

Slice Carry-Chain Timing Characteristics

Figure 31 illustrates the timing characteristics of a slice carry chain implemented in a Virtex-6 FPGA slice.

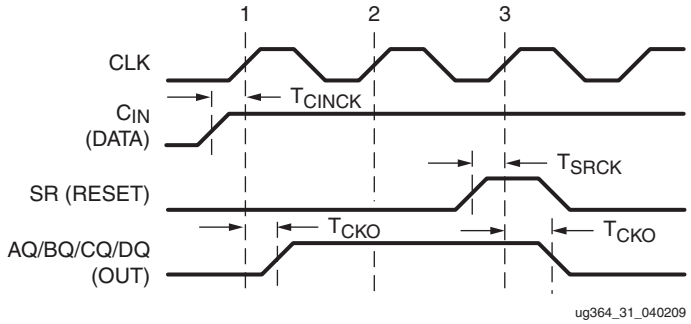

Figure 31: **Slice Carry-Chain Timing Characteristics**

- At time T_{CINCK} before clock event 1, data from CIN input becomes valid-high at the D input of the slice register. This is reflected on any of the AQ/BQ/CQ/DQ pins at time T_{CKO} after clock event 1.
- At time T_{SRCK} before clock event 3, the SR signal (configured as synchronous reset) becomes valid-high, resetting the slice register. This is reflected on any of the AQ/BQ/CQ/DQ pins at time T_{CKO} after clock event 3.

CLB Primitives

More information on the CLB primitives are available in the software libraries guide.

Distributed RAM Primitives

Seven primitives are available; from 32 x 2 bits to 256 x 1 bit. Three primitives are single-port RAM, two primitives are dual-port RAM, and two primitives are quad-port RAM, as shown in Table 10.

Table 10: **Single-Port, Dual-Port, and Quad-Port Distributed RAM**

Primitive	RAM Size	Type	Address Inputs
RAM32X1S	32-bit	Single-port	A[4:0] (read/write)
RAM32X1D	32-bit	Dual-port	A[4:0] (read/write) DPRA[4:0] (read)
RAM32M	32-bit	Quad-port	ADDRA[4:0] (read) ADDRB[4:0] (read) ADDRC[4:0] (read) ADDRD[4:0] (read/write)
RAM64X1S	64-bit	Single-port	A[5:0] (read/write)
RAM64X1D	64-bit	Dual-port	A[5:0] (read/write) DPRA[5:0] (read)
RAM64M	64-bit	Quad-port	ADDRA[5:0] (read) ADDRB[5:0] (read) ADDRC[5:0] (read) ADDRD[5:0] (read/write)
RAM128X1S	128-bit	Single-port	A[6:0] (read/write)
RAM128X1D	128-bit	Dual-port	A[6:0], (read/write) DPRA[6:0] (read)
RAM256X1S	256-bit	Single-port	A[7:0] (read/write)

The input and output data are 1-bit wide (with the exception of the 32-bit RAM).

Figure 32 shows generic single-port, dual-port, and quad-port distributed RAM primitives. The A, ADDR, and DPRA signals are address buses.

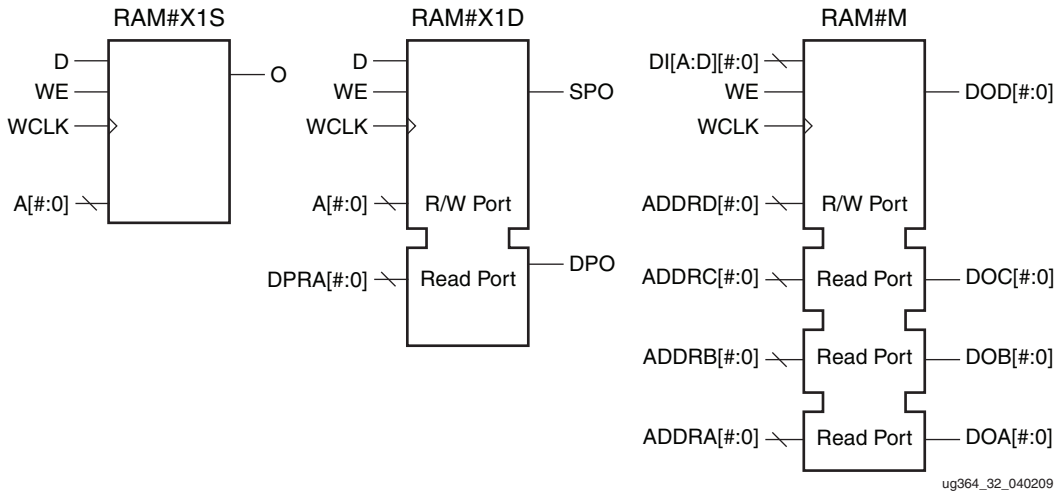

Figure 32: **Single-Port, Dual-Port, and Quad-Port Distributed RAM Primitives**

Instantiating several distributed RAM primitives can be used to implement wide memory blocks.

Port Signals

Each distributed RAM port operates independently of the other while reading the same set of memory cells.

Clock – WCLK

The clock is used for the synchronous write. The data and the address input pins have setup times referenced to the WCLK pin.

Enable – WE/WED

The enable pin affects the write functionality of the port. An active write enable prevents any writing to memory cells. An active write enable causes the clock edge to write the data input signal to the memory location pointed to by the address inputs.

Address – A[#:0], DPRA[#:0], and ADDRA[#:0] – ADDRD[#:0]

The address inputs A[#:0] (for single-port and dual-port), DPRA[#:0] (for dual-port), and ADDRA[#:0] – ADDRD[#:0] (for quad-port) select the memory cells for read or write. The width of the port determines the required address inputs. Some of the address inputs are not buses in VHDL or Verilog instantiations. Table 10 summarizes the function of each address pins.

Data In – D, DID[#:0]

The data input D (for single-port and dual-port) and DID[#:0] (for quad-port) provide the new data value to be written into the RAM.

Data Out – O, SPO, DPO and DOA[#:0] – DOD[#:0]

The data out O (single-port or SPO), DPO (dual-port), and DOA[#:0] – DOD[#:0] (quad-port) reflects the contents of the memory cells referenced by the address inputs. Following an active write clock edge, the data out (O, SPO, or DOD[#:0]) reflects the newly written data.

Inverting Clock Pins

The clock pin (CLK) has an individual inversion option. The clock signal can be active at the negative edge of the clock or the positive edge for the clock without requiring other logic resources. The default is at the positive clock edge

Global Set/Reset – GSR

The global set/reset (GSR) signal does not affect distributed RAM modules.

Shift Registers (SRLs) Primitive

One primitive is available for the 32-bit shift register (SRLC32E). Figure 33 shows the 32-bit shift register primitive.

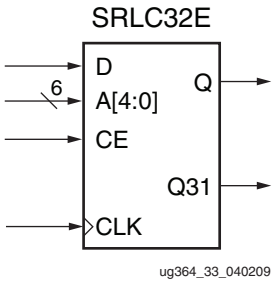

Figure 33: **32-bit Shift Register**

Instantiating several 32-bit shift register with dedicated multiplexers (F7AMUX, F7BMUX, and F8MUX) allows a cascadable shift register chain of up to 128-bit in a slice. Figure 18 through Figure 20 in the *Shift Registers (Available in SLICEM only)* section of this document illustrate the various implementation of cascadable shift registers greater than 32 bits.

Port Signals

Clock – CLK

Either the rising edge or the falling edge of the clock is used for the synchronous shift operation. The data and clock enable input pins have setup times referenced to the chosen edge of CLK.

Data In – D

The data input provides new data (one bit) to be shifted into the shift register.

Clock Enable - CE

The clock enable pin affects shift functionality. An inactive clock enable pin does not shift data into the shift register and does not write new data. Activating the clock enable allows the data in (D) to be written to the first location and all data to be shifted by one location. When available, new data appears on output pins (Q) and the cascadable output pin (Q31).

Address – A[4:0]

The address input selects the bit (range 0 to 31) to be read. The nth bit is available on the output pin (Q). Address inputs have no effect on the cascadable output pin (Q31). It is always the last bit of the shift register (bit 31).

Data Out – Q

The data output Q provides the data value (1 bit) selected by the address inputs.

Data Out – Q31 (optional)

The data output Q31 provides the last bit value of the 32-bit shift register. New data becomes available after each shift-in operation.

Inverting Clock Pins

The clock pin (CLK) has an individual inversion option. The clock signal can be active at the negative or positive edge of the clock without requiring other logic resources. The default is positive clock edge.

Global Set/Reset – GSR

The global set/reset (GSR) signal does not affect the shift registers.

Other Shift Register Applications

Synchronous Shift Registers

The shift-register primitive does not use the register available in the same slice. To implement a fully synchronous read and write shift register, output pin Q must be connected to a flip-flop. Both the shift register and the flip-flop share the same clock, as shown in Figure 34.

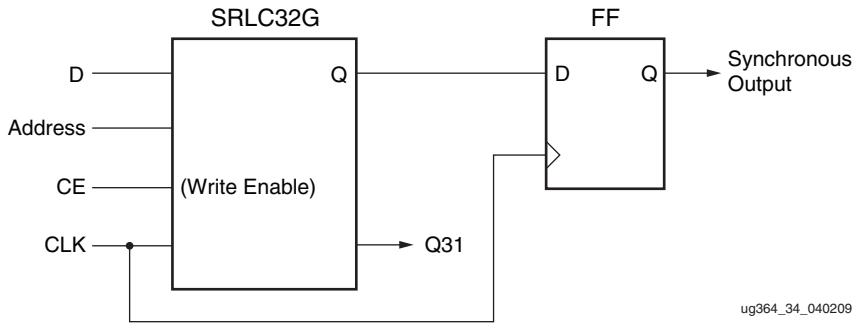

Figure 34: **Synchronous Shift Register**

This configuration provides a better timing solution and simplifies the design. Because the flip-flop must be considered to be the last register in the shift-register chain, the static or dynamic address should point to the desired length minus one. If needed, the cascadable output can also be registered in a flip-flop.

Static-Length Shift Registers

The cascadable 32-bit shift register implements any static length mode shift register without the dedicated multiplexers (F7AMUX, F7BMUX, and F8MUX). Figure 35 illustrates a 72-bit shift register. Only the last SRLC32E primitive needs to have its address inputs tied to `0b00111`. Alternatively, shift register length can be limited to 71 bits (address tied to `0b00110`) and a flip-flop can be used as the last register. (In an SRLC32E primitive, the shift register length is the address input + 1).

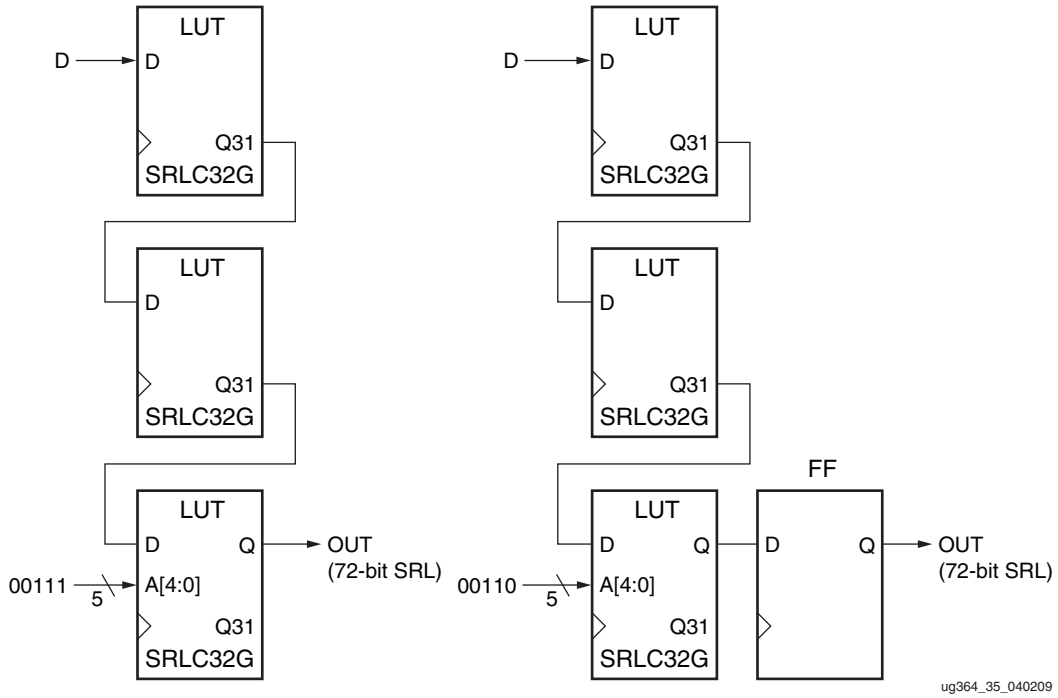

Figure 35: **Example Static-Length Shift Register**

Multiplexer Primitives

Two primitives (MUXF7 and MUXF8) are available for access to the dedicated F7AMUX, F7BMUX and F8MUX in each slice. Combined with LUTs, these multiplexer primitives are also used to build larger width multiplexers (from 8:1 to 16:1). The Designing Large Multiplexers section provides more information on building larger multiplexers.

Port Signals

Data In – I0, I1

The data input provides the data to be selected by the select signal (S).

Control In – S

The select input signal determines the data input signal to be connected to the output O. Logic 0 selects the I0 input, while logic 1 selects the I1 input.

Data Out – O

The data output O provides the data value (one bit) selected by the control inputs.

Carry Chain Primitive

The CARRY4 primitive represents the fast carry logic for a slice in the Virtex-6 architecture. This primitive works in conjunction with LUTs in order to build adders and multipliers. This primitive is generally inferred by synthesis tools from standard RTL code. The synthesis tool can identify the arithmetic and/or logic functionality that best maps to this

logic in terms of performance and area. It also automatically uses and connects this function properly. Figure 24, page 33 illustrates the CARRY4 block diagram.

Port Signals

Sum Outputs – O[3:0]

The sum outputs provide the final result of the addition/subtraction.

Carry Outputs – CO[3:0]

The carry outputs provide the carry out for each bit. A longer carry chain can be created if CO[3] is connected to CI input of another CARRY4 primitive.

Data Inputs – DI[3:0]

The data inputs are used as "generate" signals to the carry lookahead logic. The "generate" signals are sourced from LUT outputs.

Select Inputs – S[3:0]

The select inputs are used as "propagate" signals to the carry lookahead logic. The "propagate" signals are sourced from LUT outputs.

Carry Initialize – CYINIT

The carry initialize input is used to select the first bit in a carry chain. The value for this pin is either 0 (for add), 1 (for subtract), or AX input (for the dynamic first carry bit).

Carry In – CI

The carry in input is used to cascade slices to form longer carry chain. To create a longer carry chain, the CO[3] output of another CARRY4 is simply connected to this pin.